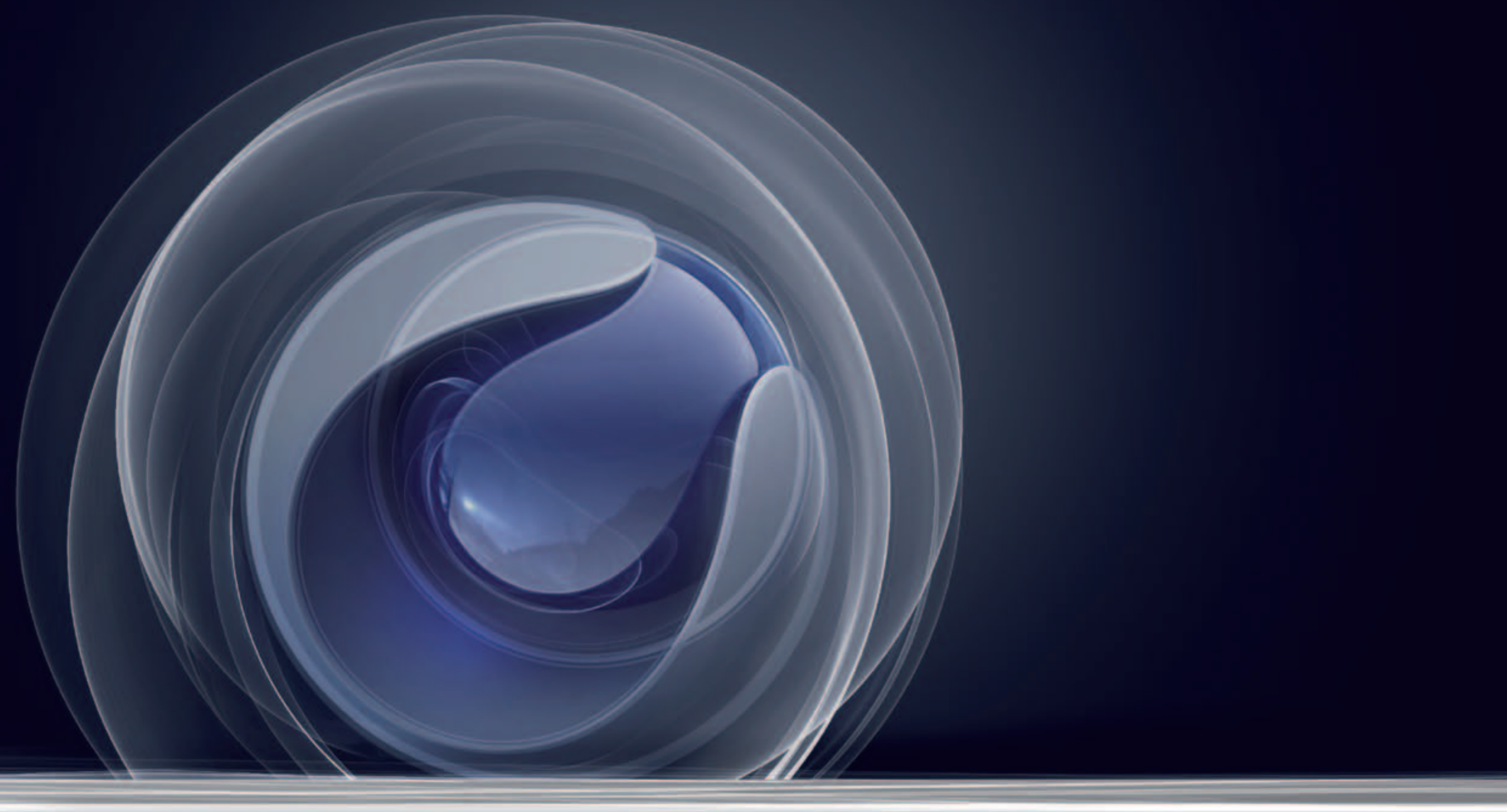


CINEMA 4D

Release 13

MAXON

Curriculum

Contents

Curriculum Description 1

Educational Philosophy of the Tutorials 1

Role of the Instructor 1

Beyond the Basics 1

About the author of the CINEMA 4D R9.x curriculum: . . . 2

About the author of the CINEMA 4D Curriculum R13: . . . 2

Macintosh Operation and Commands 3

Basic keyboard shortcuts 3

Basic CINEMA 4D Setup 4

Editor camera focal length. 4

Gouraud shading 4

Render level of detail 4

Toolbars 4

Active axis stems 4

Using the Tutorials 5

The general structure of a tutorial 5

Use of names in tutorials 5

A note about naming conventions 5

Intro to Computer Animation 6

Modeling 6

Texturing 6

Lighting 7

Animation. 7

Rendering 7

Common Icons in CINEMA 4D 9

Intro Tutorial 10

Navigation in CINEMA 4D 18

The GUI 18
Views and cameras 19
Navigating in the Viewport 19
Coordinate axes 19
Coordinate systems 20
Moving the view (camera) 20
Moving, rotating and scaling objects 20

Navigation Tutorial 22

Introduction to Modeling 31
Overview 31
Primitives 31
Parameters 31
Smoothing 32
Displaying objects 32
Level of detail 33
The Object Manager 34
Object tags 34
Object hierarchies 34
Arranging objects 35
Grouping objects 35

Modeling Intro Tutorial 36

Primitives Practice 45

Splines and Generators 49
Overview 49
Splines 49
Bézier splines 49
Useful tools and operations for points 50
Interpolation 50
A note on backing up your work: 51
Spline primitives 52
Generators 52

Caps and rounding 53

Splines and Generators Tutorial **55**

Table Lamp Tutorial **67**

Vase Tutorial **73**

Splines and Generators **76**

One generator objects 76

Multi-generator objects. 77

Polygon Modeling **78**

Overview 78

Polygons and points 78

Selecting Points/Polygons 79

Moving Points/Polygons 79

Point/Polygon modes 79

HyperNURBS objects 81

Deformers 82

Polygon Modeling Tutorial **84**

Polygon Practice **95**

HyperNURBS Bottle Tutorial **100**

Table Modeling Tutorial **103**

Pitcher Tutorial **110**

Introduction to Materials **115**

Overview 115

Materials 115

Channels 116

Textures 118

Texture map (bitmapped texture) 118

Shaders 118

3D Shaders 119
Texture mapping 119
UVW mapping .120
Tiling 120

Materials Intro Tutorial 122
Parameters in the Material Editor 125
Texture Tag Parameters 126
Diffusion 127
Bump. 128
Displacement .129
Glow 129
Transparency .129
Reflection 130
Environment .130
Luminance 131
Fog 131
Alpha. 131

Texturing and Label Tutorial 134

Table Shader Tutorial 137

Lamp Materials Tutorial 140

Introduction to Lighting 144
Overview 144
Light objects .144
Illumination .145
Shadows 146
Visible lights. 147
Other effects .147
Using lights .147

Lighting Intro Tutorial 149

Composing and Lighting a Scene 155

Transparency Tutorial **164**

Introduction to Animation **166**

Overview 166

Cameras 166

Keyframing 167

The Timeline .167

Tracks 168

Keyframes 168

FCurves 169

Animating special effects and parameters 169

Morphing and PLA 169

Animation Intro Tutorial **171**

Introduction .175

Placing keys in the Timeline 176

Keys and curves 176

Copying keys and tracks 178

Animating a deformer 179

Cameras and lights 180

Sun, Earth, Moon Tutorial **182**

Introduction to the Motion System **187**

Introduction to Rendering **191**

Overview 191

The Picture Viewer 191

Antialiasing 193

Resolution 193

Output formats 194

Other settings .194

Rendering Intro Tutorial **195**

Working with the Doodle tool **199**

Overview 201

The physics of color 201
The biology of color 201
Additive color, the RGB system 202
Subtractive color, the CYMK system 203
Color pickers 203
Overview 205
Color 205
Lighting and color 205
Reflection 206
Refraction 206
Overview 208
The camera lens 208
Focal length 208
Focal length and perspective 210
F-stop and depth of field 211
Relevance to 3D modeling and animation 211

Curriculum Description

These materials were developed for teaching computer modeling and animation to above-average high school juniors and seniors who have been successful in geometry. They should also work well for an introductory college-level course. Each major area of instruction, e.g. Polygon Modeling, contains both an introductory background section and a tutorial. The background sections are intended to be read outside of class to prepare students for the tutorials. Most units have additional tutorials to illustrate particular concepts and to offer more practice with tools and techniques.

Educational Philosophy of the Tutorials

This curriculum is structured to teach the techniques of modeling and animation in CINEMA 4D while also instilling an understanding of the underlying concepts. It describes the "why" along with the "how". Because of its methodical approach, it is more suited to a year-long class where students will have ample opportunity to execute projects in the second semester. This is a dryer approach than a high interest, project-based curriculum, but I believe it fosters more independence and versatility later on when students are cut loose to create on their own. It is recommended to occasionally break up the tutorials with other activities, as described below.

Although several of the tutorials could be used out of sequence, the basic materials are designed to be used in order. As students learn tools and commands, the detail of instruction diminishes. This is meant to promote student retention of techniques and commands. This, along with the emphasized use of standard shortcuts, is meant to increase efficiency, allowing students to be more productive.

Without a doubt, a course in computer animation will have students working at varied paces. In most of the tutorials, except for the basics, there are several optional extensions and challenges which can occupy the faster students while others catch up. For the basic tutorials I encourage quicker students to help others keep up the pace.

Role of the Instructor

While some of the more industrious and capable learners will be able to use these materials on their own, most students will benefit from instructor intervention. A word or two here and there from the instructor or a fellow student can save time and frustration.

From time to time, especially at the beginning of each major tutorial, it is advisable to demonstrate some of the techniques for the students. Even though the tutorial takes them through each step, having seen the process beforehand will help give a sense of direction to the tutorial process. Having some CINEMA 4D files prepared for use with an LCD projector is ideal.

Also, you should allow for some breaks from a straight tutorial approach. This could be a portion of Fridays for free-form exploration or perusing the MAXON tutorials for ideas and techniques. You can also introduce what appear to be spontaneous assignments from time to time. For example, you can have students model a banana with as few polygons as possible after the modeling unit has been completed. Either bring bananas in for reference or notify students a day or two ahead so they can research what a banana really looks like. As another example, after the animation unit has been completed, give students the challenge of animating the orbit of the earth, moon and sun (an exercise in using Null Object Objects to simplify rotation).

As students progress, give them more freedom to supplement their work. For instance, when assembling the indoor table scene, have them model and texture another object of their own choosing to add to the tabletop before final rendering. This also presents an opportunity to keep faster-paced students occupied with additional modeling, lighting or material tasks.

Beyond the Basics

Once students have waded through the basics you will probably want them to pursue the creation projects. A good first project might be to continue to develop the "table scene" into a full-blown interior. They may want to build a different type of table, add a couch, chair, dresser, sink, clock, water fountain (all choices of former students!), etc.

A logo project is more or less obligatory. This is fun for students because they can personalize it. An instructor can find no end to constructively criticizing the look of a photorealistic indoor scene, but how much glow is too much on a student's name flying down an ice tunnel on asteroid 707b? Logos offer the opportunity to practice animation without the structures of realism.

You should certainly allow time for students to do a creative project of their own choosing. I highly recommend requiring a brief treatment, storyboards and a timeline which will encourage planning and thoughtful development. I have also included requirements that they research and add some new techniques to each project.

About the author of the CINEMA 4D R9.x curriculum:

Byrne Pedit used his experience in the 3D graphics industry to create this useful and easily understandable C4D curriculum.

About Byrne Pedit

I have taught advanced college preparatory sciences and computer graphics at the high school level over the last eighteen years, currently in Paradise, California. Before that I had several careers, including six years in 35 mm special effects cinematography. I served in this capacity as model maker, cameraman, optical printer operator and director of photography.

I began animating to illustrate concepts for my Advanced Placement Biology students. Many of our students, partly due to these viewings, were interested in the process of 3D modeling and animation. The reasonable educational pricing, stability, feature set and integrated modeler of CINEMA 4D XL made our Computer Graphics and Animation (CGA) class a reality.

— Byrne Pedit

About the author of the CINEMA 4D Curriculum R13:

Nigel Allen has comprehensively updated and modified this curriculum to reflect R13 functionality.

About Nigel Allen

Over the past seventeen years I have worked in many areas of the multimedia industry, having been a sound engineer, freelance animator, CINEMA 4D customer support specialist (Australia). Additionally I have been teaching CINEMA 4D at many levels of proficiency in tertiary education institutions across Australia.

In 1996 I was introduced to CINEMA 4D whilst creating a computer-based multimedia training tool for an audio production company in Australia. After the project's completion I decided to make CINEMA 4D my tool of choice and have completed projects for clients including Microsoft, JVC and Toyota.

— Nigel Allen

Macintosh Operation and Commands

There are certain operations and commands which you will use very frequently. You will need to become familiar with these and, where applicable, the corresponding shortcut.

The Command (cmd) Key: This is the key with the apple and the ''propeller'' symbol. This key is used extensively on the Macintosh in combination with other keys and the mouse. When used with the click button on a Mac mouse it has the same effect as the right-click button on a PC mouse.

Basic keyboard shortcuts

You will use these commands every day. Learn them and their shortcuts.

♦ cut	Cmd+ X *(notice how X+C+V are next to each other on keyboard. Just learn ''cut/copy/paste'')*
♦ copy	Cmd+C
♦ paste	Cmd+V
♦ undo	Cmd+Z
♦ redo	Cmd+Y
♦ quit program	Cmd+Q
♦ open file	Cmd+O
♦ save file	Cmd+S
♦ close file *	Cmd+W
♦ close window *	Shift+W

**Cmd+W has the standard function of closing the active file in any Mac program. Shift+W has a slightly different function. CINEMA 4D may have multiple windows open, each containing a different aspect of the program. Since these windows always sit in front of the main window, you will need an efficient way to close them rapidly when you need access to the main window behind them. This is the function of the Shift+W shortcut.*

When using these shortcuts, use your left hand only. There are other shortcuts native to CINEMA 4D which you will learn as you encounter them later.

Basic CINEMA 4D Setup

CINEMA 4D is a highly sophisticated and capable program. It offers a wide variety of options for customizing its interface and adjusting preferences. In the following are some suggestions on how to speed up workflow, foster good habits and make a few operations easier for the tutorial process.

Editor camera focal length

CINEMA 4D's default perspective view is through a 36 mm lens. While this is a good choice for many animating scenarios, it is a bit wide for most modeling uses. I prefer setting this to 80 or even 100 depending on the size of the object. To make the change, select '*Camera*' under the *Mode* menu in the Attribute Manager. This will display the settings for CINEMA 4D Editor Camera. From here you can alter the *Focal Length* parameter. This change must be saved within a file named "new.c4d" within the folder containing the CINEMA 4D application.

Gouraud shading

I have set Gouraud Shading as the default in all views. If other modes are needed to speed up the editor or show geometry they can be changed as work progresses. These changes must also be saved in "new.c4d".

Render level of detail

From the Options window menu bar, select "Level of Detail / Use Render LOD for Editor Rendering".

Toolbars

- Remove the Undo/Redo buttons from the top tool bar. Students should use the keyboard shortcuts for these frequently used functions from the very beginning.
- Add the Camera mode to the left toolbar. This mostly adds focus for the editor camera.
- Add Render Region to the top toolbar.

Using the Tutorials

The following paragraphs describe the conventions used in the tutorials that are meant to help you distinguish between various names, terms and commands.

This tutorial is designed for use with a Macintosh computer with regard to several keyboard shortcuts.

The general structure of a tutorial

The explanatory material is written in a sans serif font.

Instructions are written in bold and colored text.

Use of names in tutorials

File and folder names: These are always within quotation marks. For example, the file "Table Lamp 1" should probably be located in the "Table Lamp" folder inside your "Student" folder.

Tools and commands: Tools, commands and the names of parameters (values which change attributes of objects) are displayed in sans serif *italics*. However, sometimes the names used in the tutorials are shortened from the exact name which appears in the status bar at the bottom of the window. For example, the word *Use* has not been used in front of tools like the "*Use* Point mode", "*Use* Object mode", etc. These have been shortened to "Point mode", "Object mode", etc.
Another example is that "*Move Active Element*" is simply referred to as the "*Move*" tool. These abbreviations are generally the terms you will eventually use to refer to these tools over time.

Object names: These are names you give to the objects you create (or that CINEMA 4D gives as default names when you first create the object). Object names will be displayed in the opposite font style as the text around them. Since the current style is sans serif, an object named here would appear serif, e.g. vase profile or vase_profile.

In the middle of this sans serif font the names would appear as vase profile **or** vase_profile**.**

A note about naming conventions

Within the tutorials, object names will often use the underscore to link separate words or link a number to a word portion of a name. Some examples might be: vase_1 or Table_Top_1. This is done to make it easier to see the name independently of the rest of the text. When you are entering names for objects it is not necessary to use the underscore. In the Object Manager, the names are in list form so there is no confusion where a name begins or ends.

For file names you need to exercise more care. For maximum portability of your files you should use only letters, numbers, the underscore and only one period (to separate the file type suffix at the end of the file name).

Intro to Computer Animation

The process of turning an idea into a finished computer animation involves several procedures regardless of the length or type of animation desired. These are described below in a sequence commonly used for animations, but most of these procedures will be used intermittently during the entire process of completing a project. For instance, it is useful to render test frames often during the texturing and lighting stages of a project, and it may be necessary to adjust lights after observing their effect on a moving object.

As you read the descriptions, keep in mind the limitations of the medium. First and foremost are the limitations of time and memory. It is easy to build a scene in five minutes that will slow your computer to a crawl or even overburden it altogether. The keyword in computer graphics is efficiency. Actually, this is not that much different from traditional movie making where a $5000 matte painting may substitute for a set which would have cost $100,000. The best way to be efficient is to thoroughly plan your project in advance.

The following is an overall survey of the process. A more in-depth look at each area will follow your introduction. There is a lot of information here which is meant to give you an overview and to convey a feeling for the process of computer animation. You will better understand these processes as we delve into hands-on experience within each area.

Modeling

Modeling is the act of creating a three-dimensional **object**. Some of these objects may serve as stationary props while other objects will be animated. There are several different methods used for modeling, depending on the outcome desired and the features of CINEMA 4D.

Some modeling techniques allow the instant creation of standard shapes such as cylinders and spheres. These are called **primitives** and can generally be created and rendered quickly. However, they are also simple in appearance and not very flexible outside of basic adjustment of scale and proportion. At the other extreme are **polygonal objects** that are made up of a mesh of **polygons**, usually quadrangles or triangles. At the corners of the polygons are the **points** which make up the most basic level of the object's structure. The data of the points' locations and how they are connected is used by the computer to generate the representation of the object we see on the screen.

Polygon objects offer control of surface geometry at the level of individual polygons or points. More polygons can be used for a finer and more detailed geometry, but high polygon counts are usually the number one factor in slowing the computer system down.

There are many tools for building and modifying models. Some of these work on the model as a whole. The **Deformers**, for example, can bend, twist and shatter objects, Among other functions. Other tools work on individual polygons and points. It is important to know the role of each object in the finished product before you begin. If an object is to appear in the distant background it would be a waste of time and system performance to model it with a polygon count of 50,000. On the other hand, a featured object in a close-up will look faceted and crude if not modeled in sufficient detail. Objects whose shapes are to be animated require special attention and planning.

Texturing

Once modeled, the surface of the object needs to be covered with a **material**. Once a material is applied to an object it is usually called a **texture,** and the process of applying the material is termed **texturing**. Most materials have different properties such as color, reflectivity, transparency, shininess and relief (bump), each of which is termed a **channel**. To make these materials more interesting or appropriate to the object, a two-dimensional image such as a photograph may be used in almost any of the channels. When used in this way, these images are called **texture maps**.

Placing a two-dimensional texture onto a 3D object is called **mapping**. Except for the simple case of *flat mapping* a label onto a box most mapping results in distortion in various regions of the object, such as the puckering which appears at the poles of a sphere when *spherically mapping* a texture onto it. Work, skill and specialized painting programs, such as MAXON's BodyPaint 3D can help minimize these distortions.

Mathematical formulae can also be used to generate surface textures. These are known as **procedural shaders**, or simply **shaders**, since a certain mathematical procedure is used to calculate the texture.

With a *3D procedural shader*, or **volume shader**, the texture is generated in three dimensions with the object's surface defining where it will be seen. There are no seams or distortion with these 3D shaders but they will take longer to render.

Lighting

Lighting is very important in setting the tone for a scene as well as bringing out the surface look you want from your objects. Most scenes will have a principal light called the **key light** that will not only be the brightest but will generally be the only light casting dominant shadows. Other lights called **fill lights** are used to reveal other surfaces in the scene or to simulate the lighting in natural scenes as it is bounced off surfaces or scattered by the atmosphere.

Scenes may also have **practical lights** that are objects that would normally generate their own light, such as lamps, headlights or fireflies.

There are several different light types. The **infinite light** shines equally in all directions and appears to come from an infinite distance, e.g. as the sun does. The **spot light** is contained within a cone which can be aimed toward a selected part of the scene. An **omni light** shines outwards in all directions from a point, like a bare bulb. Many different properties of lights can be controlled and animated. Lights can vary in intensity, fade out over a distance (**fall-off**), cast various types of shadows, be made visible, as if shining through smoke, and much more.

Animation

The models are brought to life through animation. At its simplest, this involves changing the position, rotation or scale of an object over time. For more complex tasks such as **character animation**, where movement like that of living creatures is achieved, various deformations in the model itself are programmed over time.

The principal mode of achieving movement over time is through **keyframe animation**. Here, the object is recorded in its present state at a given instant in time (a keyframe). Then a different frame at a later time along the timeline is chosen, the model is manipulated to a new state or position, and a new keyframe is recorded. CINEMA 4D then calculates the motion between keyframes. Most quality programs allow you to view and edit the path which the computer has calculated through your keyframes.

Animated movement may also occur within the object itself. This is called **morphing** if it is done by using two or more separate models and having CINEMA 4D calculate the intermediate shapes between. Most applications allow for **point level animation** in which positions of individual points within an object are moved and then keyframed over time.

Animation may also be created by means of deformers (already mentioned above under "Modeling"). The most useful and versatile of these is **bones**. Bones can influence an object in various ways. For example, bones can move an arm or be used to move lips to create a smile.

Almost all characteristics of models, textures and lights can be animated. And, of course, the camera can also be animated.

Rendering

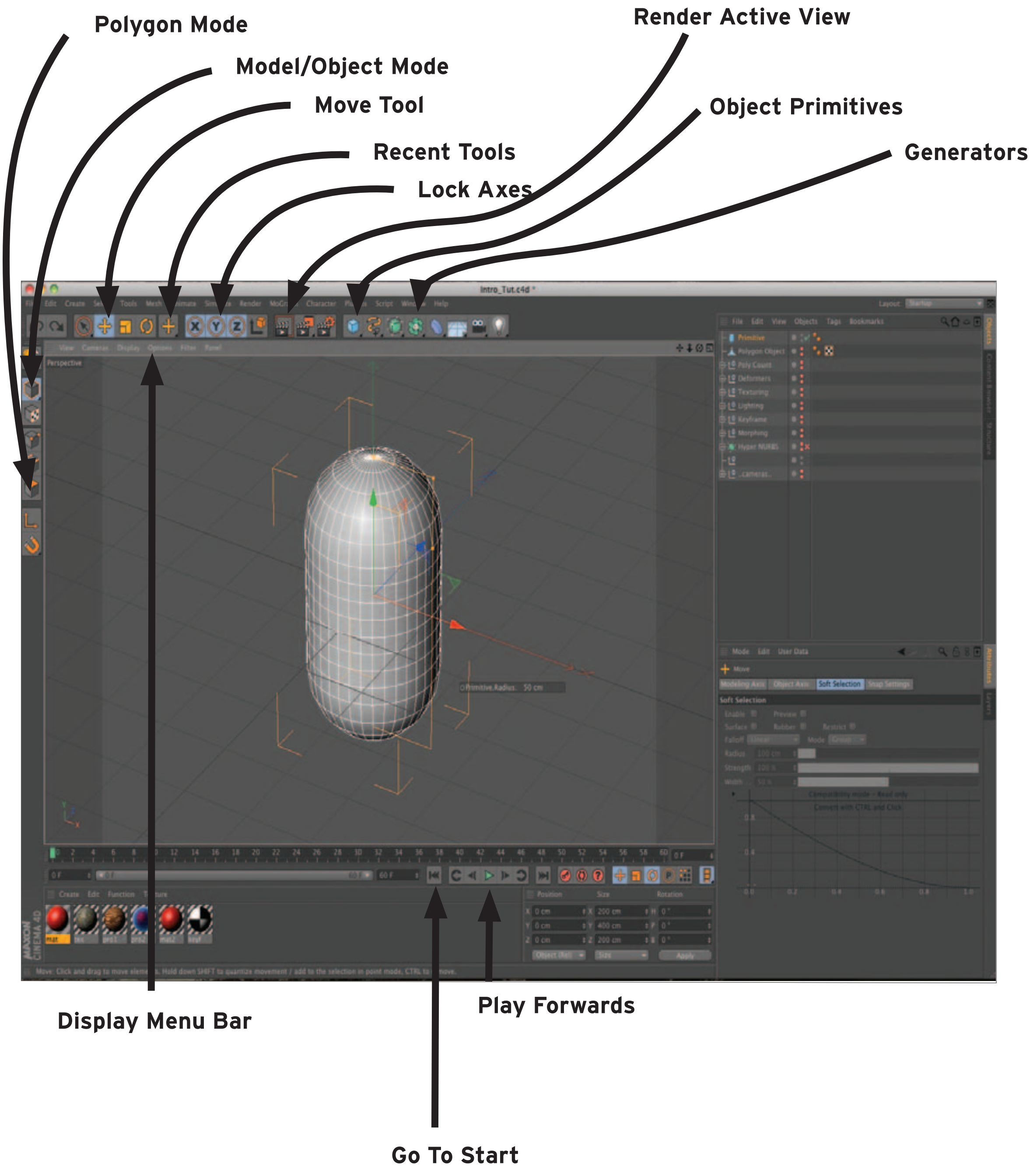
The final visual evaluation of all data and output in a desired form is called rendering. This might be a still frame at low resolution for use on the web, a high resolution still for printing or a movie in one of many formats.

High-end software packages use a rendering technique called **ray tracing**. This process follows rays of light back from the camera as they bounce off the objects' surfaces from the light source. Raytracing allows the display of reflections and refractions of light interacting with surfaces.

During rendering, various effects such as motion blur and fog can be added. Rendering is also where a process called **antialiasing** is used to take out the "jagged edges" or stair-stepping effect that arises due to the fact that the screen image is composed of small squares of color (pixels).

The rendering stage is also where you set the **resolution**. This determines the size of your rendered output, whether for an 8x10-inch print, a poster, a web image or video to be output to TV or film.

Overview of the CINEMA 4D GUI



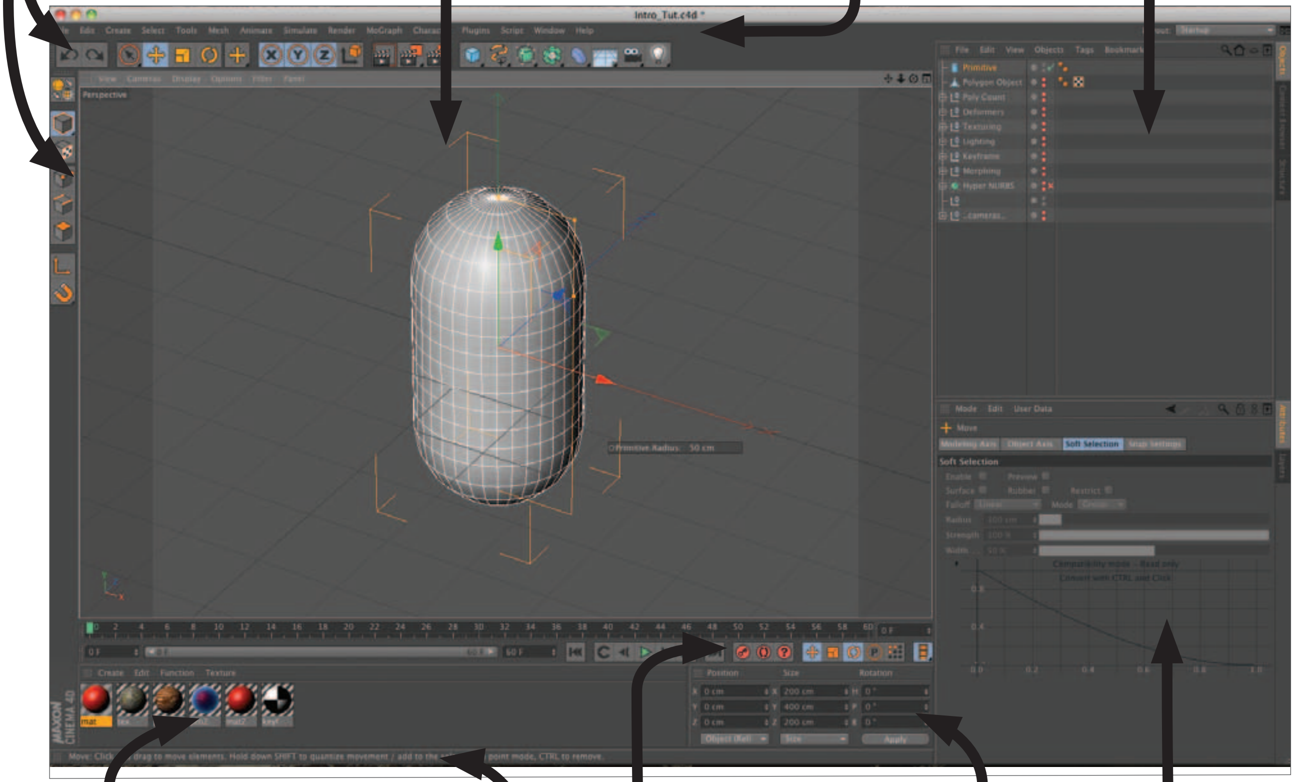
Common Icons in CINEMA 4D

Viewport: Can be configured to show up to four different views from any chosen perspective. (Also called the view or the edtor).

Toolbars: (tool/Command Pallets) Quick access to commonly used tools and commands.

Main menu: Allows access to most functions and commands. Many of these are also available from the Command Pallets.

Object Manager: Here you select, organize and assign various attributes to the objects you have created.



Material Manager: This is where you create, store and access the materials used in the scene.

Animation Toolbar: Allows viewing animations over time and recording of keyframes for animations.

Attribute Manager: Here you adjust all the available options and parameters for a selected object, tag or tool.

Coordinates Manager: Displays numerical info about an object. You can precisely position, scale and rotate objects here.

Status Bar: Describes the name and functionof various icons as you move the cursor over them. Also shows rendering time and progress.

Intro Tutorial

This tutorial follows the "Intro to Computer Animation" handout that introduces the process of computer animation. You should have read the handout by now and should have it handy for review as you go through the tutorial. It pretty much follows the order in the handout, focusing on the terms in bold-face type. Go through this tutorial in order and pay attention to the commands. As you progress, the instructions for operations you used previously used will become less frequent. You are not just learning about CINEMA 4D and animation in general, but also practicing the commands and operations you will need to use like a second language if you want to be productive.

1. Opening CINEMA 4D

Open the CINEMA 4D program by double-clicking on its name or icon in the "CGA" folder. To maximize use of the screen, follow the directions in the Basic CINEMA 4D Setup paper.

2. Opening the file

Open the tutorial "Intro_Tut.c4d" using the keyboard shortcut Cmd+O. This file is located in the "Intro Tutorials" folder within the "CGA" folder.

You also could have simply opened the "Intro Tutorials" folder from the desktop and double-clicked on the "Intro_Tut.c4d" filename. This would have both started CINEMA 4D and opened the file.

3. Showing and hiding objects

Study the Object Manager. In the left column are the names of the objects in the scene. A box with a "+" on the far left means that the object has sub-objects (children) and can be expanded to show those Children by clicking on the "+" box. Try it. Note how the sign in the box changes to a "-", indicating the Parent object is expanded. Click again on the minus box to collapse the Children into the Parent. This makes the Object Manager less cluttered and easier to navigate. The first icon to the right of the name shows the object type.

In the narrow column down the center of the Object Manager are three buttons for each object. The larger button controls the assignment of the object to a layer. The two smaller buttons represent the following display functions: The top button controls whether or not the object is displayed in the editor (what you see while working on modeling an object or animating a scene). The bottom button controls whether the object is displayed during rendering. The color code is:

- Red: Always off
- Green: Always on
- Gray: On or off depending on the Parent

Gray is the normal mode for a displayed object.

The check mark or X to the right of the buttons controls whether a deformer or parametric object is active or not.

The column at the right holds icons representing "tags" that are used to assign certain properties to the object.

Note: You will note that all the display buttons for objects at the top-most level (parents) are red, except for two (currently, only parents are shown in the list). The object with no name (second from the bottom) is a Null Object (an empty object with no data). Null Objects have many important uses but the one used here is simply for spacing, separating the cameras from the other objects. The other object with gray buttons is the one titled Primitive, which is visible in the viewport.

Click twice on the top button associated with the object Primitive.

The button should turn red and the object will disappear from the viewport.

Click once more on the same button.

Now the button has returned to gray and you can again see the object in the editor window. A shortcut for clicking twice on the button is to Shift+click once. Later on well see how the green button is used.

4. Primitive objects and parameters

The object Primitive is an example of a primitive object. It is not made up of polygons and points (at least not directly) but is generated by a mathematical formula in the program. There are certain values which you can set which control the shape, proportions and other features of a primitive. These modifiable characteristics are called parameters.

Make the object Primitive active by clicking on its name or the icon immediately to the right of its name.

The name should turn orange to show the object is active. Now that the object is active, its parameters can be viewed and adjusted in the Attribute Manager. The Attribute Manager will show a list of parameter group tabs laid out horizontally near the top. For this object, the groups should read Basic, Coord., Object, Caps, Slice.

Click on the Object group tab. Shift+click on the Caps group tab.

If you want to hide a group in order to reduce clutter, use *Right-click* to alternately show or hide a tab). You can scroll the Attribute Manager when the list of parameters is too long to display in the space provided. You can also make the Attribute Manager window larger.

Position the cursor on the line dividing the bottom of the Object Manager window from the top of the Attribute Manager. When the cursor turns into a double arrow, click and drag upward.

This works in many other locations around the CINEMA 4D interface as well. Now let’s adjust some parameters.

Double-click in the input box to the right of the parameter listed as Radius. Enter the value 100 and press “return” or “enter”.

Performing almost any other action would have also had the effect of activating your new entry. Notice the change in the object.

Now restore the former value by performing an Undo (Cmd+Z).

Here is a second method of adjusting a parameter.

Carefully place the cursor over the little square button with the tiny up and down arrowheads (triangles) at the right of the input box. Click and drag upward, then downward. Watch the object in the viewport change dynamically as you drag.

You may have noticed a change in the rounding at the ends if you made the object thin and then expanded it outward again. This rounding (referred to as *Fillet* in CINEMA 4D) cannot be greater than the radius of the object. Therefore it was automatically reduced as you reduced the *Radius*.

Use *Cmd+Z* to restore the primitive to its original settings.

Now for a third method of adjusting some parameters directly in the editor window (viewport).

Click on the orange point an the top right and drag to the right.

Notice that the diameter of the object changes dynamically. You can also see in the Attribute Manager how the *Radius* of the object has changed from its initial value of 50 to wherever you have moved it.

There is yet one more method of adjusting parameters directly in the viewport.

Undo the previous change. In the light blue box in the viewport reading “Primitive Radius”, click on the number 50 and drag to the right.

This box is an example of CINEMA 4D’s Heads-Up Display (HUD). You can set up one of these boxes for most light, object and material parameters, among others.

Take a minute to experiment with some of the other parameter adjustments, both dynamically in the editor and using the dialog box via the Object Manager (try, for instance, changing *Rotation Segments* to 3. Also, change the *Segments* ; also, set the Segments value in the Caps Tab's Fillet menu to 1).

So, what's a parameter again? Think of the cylinder primitive as a cylinder-making machine (after all, that is what it is). The machine is flexible (within the limits of cylindrical shapes) and needs certain information specifying how the cylinder should be formed, such as height, width, how rounded the edges should be and so on. The values that you enter into the machine (via the Attribute Manager) that specify the cylinder's characteristics are called "parameters".

5. Polygon Objects

Double-click (or Shift+click) both visibility buttons of the object Primitive. This will hide the object in the editor (as well as for rendering). Single click on both buttons of the next object, Polygon Object to display it.

Here we will take a look at a polygon object. So far it should look the same as the primitive Primitive. However, notice that the icon is different.

Click on the object's name or icon to make it active. Study the options available in the Attribute Manager.

Notice the only group tabs shown in the Attribute Manager are Basic, Coord and Phong. The Basic group, which is available for any object, lets you change the object's name and other basics such as whether or not the object should be visible in the editor (this is exactly the same as using the buttons in the Object Manager, except the visibility can be animated via the Attribute Manager, as we will see later in the course). The other group tab in the Attribute Manager is the Coordinates Tab. This simply displays and lets you set the position, scale and rotation of the object (same as using the Coordinates Manager).

The major difference you should note, compared to the primitive object, is that there are no parameters to adjust (unless you want to loosely consider position, scale and rotation as parameters).

Now, select the Polygon Mode from the Command Palette (look at the status bar as you drag the mouse over the tool icons to help you choose the proper tool. The full name for this command is Use Polygon Mode.)

Now you can see that this model is composed of many polygons (in this case they are all four-sided or rectangular).

Be sure the Move Tool is selected. Click once on one of the polygons. It will turn orange when selected. Now click and drag anywhere within the editor window.

Notice how the polygon follows your movements. You cannot do this with a primitive object without using a special object called a deformer.

"Undo" to return to the initial state. Now, enter "Point" mode which is located just above the Edge Mode. Select and move a single point as you did with the polygon above. This mode will allow finer control over modeling and animation. Make sure to Undo each change you make.

You are probably thinking that polygon and point manipulation leaves the object pretty jagged-looking - too angular for most uses. As you will eventually see though, there are ways to work with polygons to make them quite smooth.

6. Polygon counts

Hide the object Polygon Object and show the object Polygon Count by clicking accordingly on the buttons in the Object Manager. Unfold (expand) Polygon Count by clicking on the "+" box (just left of its name) to display its three Children. Select the Polygon mode. Now select each of the sub-objects (cylinder 1, etc). in order.

Notice how the polygon count changes from left to right. Notice also how the higher polygon count is associated with the smoothest object. However, there is a price to pay for this smoothness.

Select Cylinder 1 in the Object Manager. Get info on this object by selecting Object Information from the Object Manager's Objects menu. (this means: select the *Objects* item from the menu in the Object Manager. When the *Objects* sub-menu comes up, select Object Information. This is how menu selections will be shown from now on).

Notice the object has 64 polygons which take up only about 2 kB (kilobytes) of memory.

Click OK or hit return to close the info box. Now get info on object Cylinder 3.

More polygons, smoother, but now using 30 kB of memory.

The increase in memory usage of higher polygon count objects is also associated with slower redraw times as you are working in the editor, and results in longer render times. Over-use of polygons can slow the system to a crawl. One of the challenges to a successful animator is to use the minimum number of polygons necessary to achieve an acceptable image. An object at a large distance from the camera or one quickly moving across the screen does not need the detail or attention which a five-second close-up on a face would require.

7. Deformers

Fold (collapse) Polygon Count by clicking on the "-" box next to its name, then hide the object. Show (make visible) and unfold (expand) Deformers. Switch to cam 2 for viewing the scene (editor menu: Cameras / Use Camera / defCam).

Deformers are used to apply parametric deformations to primitive and polygon objects.

Click on the X in the center column of the Object Manager associated with the Twist **Deformer to activate the Deformer.**

Observe how the object changes. You can adjust the parameters of deformers just as with parametric objects.

Select the Twist **object to bring up its parameters in the Attribute Manager. Change** *Angle* **to 100 and click "OK".**

Note the change in the twist of the object. Deformers can also be adjusted interactively.

Undo. Make sure the "Model mode" is active. In the viewport, grab the orange point at the end of the orange line and drag.

Looking good!

Undo. Now disable Twist by clicking on the check mark and enable Bend. Play around with its parameters a bit but finish with an Undo for each change you make.

Multiple deformers can be used on the same object.

Now try activating both Twist **and** Bend **Deformers together.**

When multiple deformers are used together, the order in which they are placed does matter.

While both Twist and Bend are still activated, select Bend by clicking on its name and drag it upward to just above Twist. Notice a little box appears with an arrow under it while you are dragging. Drag Bend above Twist (but below basic object) until the arrow points to the left (if you were to release the mouse button while the arrow was pointing down, Bend would become a Child of Twist. However, this is not what you want in this case).

Note that the object now appears different due to the different as a result of the different order of the Deformers.

Return Bend **to its former position below** Twist**.**

Disable Twist **and** Bend**. Enable** Explosion**. Grab and move the interactive handle in the editor window.**

This deformer, like all deformers, can also be animated.

Return all deformers to their former state and disable them. Fold and hide Deformers**.**

8. Materials and textures

Unfold **Texturing.**

Notice that the Child object Material has gray visibility buttons but nothing is visible in the editor (viewport). This is because its parent object (texturing) has been hidden, as the red buttons show.

Click once on the visibility buttons of **Material.**

The buttons should turn green, overriding the red of the Parent and making the Material object visible.

All the five sub-objects (children of Texturing) are identical except in the material used. Materials are used to cover objects and give them various looks.

Double-click on the material "mat" in the Material Manager.

The Material Editor window should open up allowing editing of the material's properties (materials can also be edited in the Attribute Manager but the Material Editor is better for most purposes and will, for the most part, be used these tutorials). If the Material Editor is taking up too much space on your screen you can drag it to a new location by clicking and dragging on its title bar.

Notice the list of channels down the left side of the Material Editor window. Only two are being used for this material.

Click on the check mark next to **Specular.**

You have just disabled this channel. Note how the highlight on the sphere in the Material Editor's preview window of the Material Editor as well as that on the object in the viewport disappears.

Enable **Specular** again by clicking in the box next to the left of **Specular.**

We will now make an adjustment to the color of this material.

Click on the text **Color** (not the check box) to return to the color editing channel. Click on the button below the color preview and select "HSV".

If you click on the check box you will disable the color channel. You should now see four sliders next to the color patch in the right pane of the Material Editor (if not, click at the right-facing tiny arrowhead between the text "Color" and the small red color patch). The top rainbow-colored slider is labeled H for hue.

Drag the white triangle just under the H slider anywhere to the right and release the mouse button.

You should see the new color on the object in the viewport as well as the preview icon in the Material Editor. Now try adjusting the saturation (s).

Drag the white triangle just under the S slider anywhere to the left and let up and release the mouse button. Try adjusting the **Brightness** as well.

The "V" slider controls value which, for most practical purposes, is the same as brightness.

Hide the object **Material** and show the object **(green).**

This material uses a texture map in its color channel.

Open the material **"tex".**

In the color channel you will see the texture map along with its name (alskin01.jpg). The file for this texture map must be available on the computer for CINEMA 4D to access. If you have a small screen, either drag the Material Editor down so it is mostly off-screen or close it to have a full view of the viewport.

Render the object by clicking on the **Render Active View** icon.

You have just created your first render. Looks a bit better, does not it? The pinching of the texture at the ends of the object is a necessary artifact of trying to apply a flat, two-dimensional image to a three-dimensional surface. More on this later.

Hide Texture and show Procedural 1. Render the active view.

Rendering takes longer now and the texture is not pinched at the end. This is not a flat texture map stretched over the surface. It is a three-dimensional calculation created by applying certain mathematical procedures (hence the name procedural shader). This one is wood. If we were to cut a piece out of the object we would see the texture just as it would look if this were a real, solid piece of wood with a piece cut out of it. To see an example of this:

Hide Procedural 1 and show Procedural 1b, then render.

The price you pay for this fit is the slow render time you experienced.

Hide Procedural 1b and show Procedural 2. Render.

This is another example of a procedural shader. Procedural shaders can provide many different looks. They are also adjustable, but not in the same way as the materials with their fixed set of channels. Each procedural shader has its own dialog and its own adjustable parameters. A few adjustments to this one and you could make the object look like something you would not want to touch.

Hide Procedural 2 and show Material 2. Render.

This is an example of a material, created with the CINEMA 4D Material Editor, which uses a 3D procedural in one of its channels (*bump*). Notice there is no distortion at the ends, as was the case with the flat texture map.

9. Lighting

Hide Material 2 and fold Texturing. Show and unfold Lighting.

Good lighting is necessary to bring out the color and geometry of your objects. Until now you have been viewing objects with only the AutoLight. As soon as a Light Object is created and activated the AutoLight is switched off and the scene is lit by the active lights which you have created. Currently, only the key (Key), or principal light is active and the face appears harshly lit. In the real world, other lighting, such as that scattered by the sky or bounced off walls, would contribute to illuminating the object.

Switch on the light Fill 1 by making the object visible.

A slight blue light is emitting from a clear sky.

Now switch on Fill 2.

Here some light is reflected, perhaps by the floor. Notice how you can see some of the light object itself in the editor window. The other light objects are currently outside the viewing frame.

Lighting can become one of the most demanding aspects of a scene, especially if you are trying to achieve a realistic look. There are many more uses of lights which we will discuss in detail later.

10. Keyframe animation

Hide and fold Lighting. Show Keyframe. Select camera 3 from the Cameras menu: Cameras / Use Camera / keyFrame Cam.

Keyframe animation is the most basic technique of animation. First, view the animation.

Click on the Play Forwards button in the Animation Toolbar.

The animation should keep cycling in the forward direction.

Stop the animation by clicking on the Play Forwards button again, which has now become the Stop Playback button.

We will now take a look at keyframes:

Open the Timeline using Shift+F3 (or from the main menu: Window / Timeline). (your computer may also be set up to use the shortcut Shift+E).

Names of objects are to the left (only some are shown here) and the little rectangular blocks to the right are the keyframes.

At the top of the Timeline is the Timeline Ruler along which the frame numbers are shown (or time in seconds depending on your preference settings). Immediately below the Timeline Ruler is the Summary Track. This track enables us to see the location of all animation keyframes in the current project in a single track. To move the view to another time you can click on the desired position along the ruler (in line with the displayed numbers) or move the time marker (the green rectangle in line with the frame numbers).

Grab the time marker and move it back and forth (this is called "scrubbing"), especially across frame 20. Observe the animation in the viewport.

Note how frame 20 seems to be special. It is the point where the motion goes from falling to rolling. This is due to the keyframes positioned at frame 20.

Close the Timeline window using Shift+W (when using this command, always be sure the Timeline is the active window. That is, make sure its title bar is not grayed. Otherwise you will close the wrong window). Stop the animation if it is still going. Return to frame 0 by clicking on the Go to Start button.

11. Morphing

Hide Keyframe and show Morphing. View the scene through camera morph Cam.

This is a demonstration of how an object's shape may appear to be animated by having CINEMA 4D calculate different intermediate positions between two models with different shapes. This is called morphing.

Play the animation forward. Stop when you have had enough fun.

12. HyperNURBS

Hide Morphing. Show and unfold HyperNURBS. Return to using the first scene camera (intro Cam) for viewing the scene. Activate the Polygon mode and select the object mesh.

This is a basic polygon object. One of the most powerful functions CINEMA 4D offers is the ability to use polygons and points like a framework from which to suspend the surface, offering ways to model curved organic shapes. This is the HyperNURBS tool. To see it in action:

Activate the HyperNURBS Generator, clicking on the X next to HyperNURBS's icon.

Notice how the surface no longer fits the polygon geometry. However, it is still controlled by the points at the corners of the polygons. This is CINEMA 4D's most important tool, especially for modeling organic shapes.

13. Rendering

Rendering is the last stage at which the look of your scene can be defined. Various finishing touches can be added during rendering, but one of the most important is antialiasing. This involves removing the "jagged edges" that are a result of displaying images with a fixed number of small, visible squares (pixels). Aliasing is most evident where straight lines run at angles across the screen.

Keep HyperNURBS visible but switch off the HyperNURBS Generator itself so only the polygon geometry is displayed. Set Antialiasing to None in the Render Settings menu. Render in the active window.

Notice the staircase effect ("jagged edges"), especially at the mouth of the bottle.

Set Antialiasing back to Geometry and render the scene.

Notice the difference in appearance of the edges. Also note differences in the render time, as shown in the status bar.

There is always a price to pay for perfection, usually in terms of time.

This concludes the tutorial. We will investigate all of these functions again in much more detail later.

Close the file. If you are asked whether you want to save the file, select No.

Navigation in CINEMA 4D

The GUI

You have already been introduced to the CINEMA 4D graphical user interface, or GUI, through the "Overview " diagram. In the following are more details on the various components and features.

Managers

The functions of a complicated and versatile program such as CINEMA 4D are divided into a number of modules, or program elements. Each is accessed through its own window. The most important managers are briefly outlined below under "Windows".

Menu bars: At the top of each manager's window is a menu bar which gives access to commands and functions are particular to that manager.

Context menus: A window's menu can be quickly accessed by right-clicking anywhere in the window.

Windows

All of the program's features are displayed and accessed in windows (or through toolbars which offer direct access to the tools available in the menus of different windows). The most important of these windows are:

Viewport: This term refers to a single window through which you view the scene. It may also be referred to as the view, the editor window or simply the editor.

View panel: This term generally describes the arrangement of several viewports displayed simultaneously, such as the four-view mode of perspective, top, side and front views. If the panel displays only one viewport, then the terms view panel and viewport essentially mean the same thing.

Object Manager: This is where you select, arrange and assign attributes to the objects in a scene.

Attribute Manager : This is where you manipulate the available options and parameters of the selected (active) object, texture, material, keyframe or any other item which has user-adjustable input. This also includes any tool with adjustable parameters which has just been selected. You can also animate parameters from within the Attribute Manager.

Material manager: All of the materials in your scene are stored here.

Material Editor: Various attributes of materials and textures can be defined, adjusted and animated in this window. Materials can also be modified in the Attribute Manager but the Material Editor is better suited for these purposes.

Timeline: Various aspects of animations such as keyframes and timing of objects' movements are displayed and controlled here.

Coordinates Manager: Here you can enter numerical values to very precisely control position, rotation and scale of objects. You can also do this in the Attribute Manager.

Window styles:

Windows may have different appearances depending upon your preference and their use. The three styles are:

Free (or floating): These have the appearance of normal file or folder windows with a title bar at the top. They are movable on the screen but always lie on top of the main application window. They may be closed at any time. The two principal windows displayed this way are the Material Editor and the Timeline .

Docked: These windows are fixed in position within the main application window. This allows you constant access and more efficiently uses space by reducing borders. Examples include the View Panel and the Material Manager.

Tabbed: Tabbed windows are also docked but must be selected by clicking on a tab at the right edge of a Manager window. This allows multiple windows to occupy the same space in the main window, although only the active one is shown. The Object, Content Browser and Structure windows are examples of tabbed windows.

All of these styles are interchangeable. A tabbed window can be made free, a free window can be docked, and so on.

Views and cameras

The view panel can be arranged to show up to four different views (viewports) at once. Usually you will use a single viewport to increase redraw speed and offer a more comprehensive look at the scene. Since objects are constructed and animated within the view window, it is also referred to as the editor window.

Each view will initially be displayed by an **editor camera** whose state remains unchanged when switching between views, i.e. if you switch to another viewport and then return. The display will appear as you left it, unless you switch to another camera within that viewport. More permanent and animatable camera views are created with **scene cameras** . You may define any number of scene cameras and can lock them to preserve their settings.

There are several types of fixed camera views available from the Cameras menu in the view panel window. The left, right, front, back, top and bottom views all offer a view without any perspective (orthographic). This means that an object 100 units wide and 1000 units away appears the same size on the screen as a 100 unit wide object only 1 unit away. This is unlike the real world where more distant objects appear diminished in size (perspective). However, these so-called **parallel** views allow you to precisely position and align objects and offer more control in the selection of points. The other type of view is **perspective** and causes more distant objects to appear smaller, just as with our eyes viewing the World. A perspective view has an adjustable focal length associated with it just like a real camera lens. The perspective view lets you view the scene from any angle and distance with a perspective (focal length) ranging from extreme fish-eye to extreme telephoto (which begins to look like a parallel view). You will almost always use a perspective view to render a scene in its final form.

At the right side of each viewport menu bar are four icons which can be used to manipulate the viewport. However, you should quickly learn to use the keyboard shortcuts for these frequently used functions. The function keys *F1* through *F5 will each display a different view in the viewport* (do **not** try to toggle views using the *Cameras* menu. You will risk changing the camera in that view, thereby loosing any positional information). The other shortcuts are outlined below under "Moving the View".

Finally, *Cmd+Z* only works on changes involving objects, materials, animations etc., not on changes to the view (or camera) itself. Changes made to the view in an editor window can be undone by pressing *Shift+Cmd+Z* (however, if you select and move the camera object itself, undo with *Cmd+Z* like with any other object).

Navigating in the Viewport

As you make changes to the view you should know through which camera you are viewing the scene. If you are viewing through a scene camera you will lose your previous view angle. However, you can return to the previous angle by pressing *Shift+Cmd+Z* if you have not done too many other operations since. Generally, you should use the editor camera if you wish to move the view when modeling objects or setting up the scene. You can also create additional cameras and lock them (using a *Protection Tag*, which can be found in the Object Manager for the purpose of returning to a set, non-standard view.

Coordinate axes

The three dimensions are defined as follows in CINEMA 4D, as seen from the default perspective view:

- **X-axis**: Left to right (positive direction); color = red
- **Y-axis**: Bottom to top (positive direction); color = green
- **Z-axis**: Front to back (positive direction); color = blue

For example, a top view would look down along the Y-axis and show the Z-axis pointing up and the X-axis pointing to the right. In the standard perspective view the back-to-front axis (negative Z direction) points toward the left and down. If you are accustomed to drafting, calculus, other 3D or engineering applications you will notice that they use a different system. However, do not worry, will soon get used to the CINEMA 4D system.

Coordinate systems

World Coordinate System: These axes are always fixed in space. This system has an origin (X=0, Y=0, Z=0) which is always the fixed center of the CINEMA 4D universe (any apparent movement of the origin will be a result of a camera move). It is also from this system that the standard camera views (front, right, etc.) are defined.

Object coordinate system: These axes are associated with individual objects and may be moved and rotated as necessary, especially for the purpose of making animation and modeling easier.

Moving the view (camera)

• **Translation** (position or move): To move the view (camera) left, right, up or down, hold down the "1" key and click and drag in the view window. To move the view (camera) toward or away from the scene, simultaneously press the Control key with the 1-key, then click and drag. The in and out movements of the camera are achieved by using the Ctrl-key and will be perpendicular to the current view. You can keep the 1-key pressed or release the Ctrl-key as needed (holding down the 2-key (without Ctrl) while dragging has the same effect as the combination Ctrl+*1*). You will also notice that the focus of interest for the camera when translating in and out of a scene is set by the location of your initial click prior to dragging the mouse. This is also confirmed by a small grey cross in the editor window.

• **Rotation**: To rotate the view, hold down the 3 key while clicking and dragging. The center of rotation for these operations is determined by the active object, even though the entire scene will be rotated. This allows for maximum flexibility (if no object is selected, the scene will rotate around the World origin). Holding down the *Cmd*-key with the 3 key will force the rotation to be circular on the plane of the screen.

• **Zooming**: To zoom the view, press Ctrl+*2* while clicking and dragging. However, you will rarely want to do this! Zooming the view will change the focal length of the current camera, which will affect the perspective of how distant objects appear relative to close objects. You should use the translation movement for moving in and out since this actually moves the camera in and out. The exception is when you are in a parallel view where the camera distance is not relevant. Here, even using the Move Tool (translation), the view will be zoomed.

Remember, translations made to the view (or camera) can be undone with *Shift+Cmd+Z* (the icons in the upper right of the viewport can also be used to navigate in the view, but you should refrain from using them since they will slow your workflow).

The Camera mode

When the Camera mode is selected, dragging in the editor window will cause the view to translate, rotate or zoom, depending on whether the *move*, *rotate* or *Scale* Tool is selected. The results will be the same as when using the 1, 2, or 3 keys as described above. Again, when you move the view you are in essence moving the camera (except for scaling, which changes focus). One difference in moving views in this manner is that you may lock individual axes, thereby restraining movement to a desired direction (see below). When using the Camera mode you will also see the camera's position and focus dynamically updated in the Coordinates Manager as you move the view (camera) around.

Moving, rotating and scaling objects

Objects may also be translated, rotated or scaled just as views can be. However, there are some complications here so read the following carefully. Unlike views, objects may be moved according to either the objects' own coordinate system or the World (global) Coordinate System.

Locking axes : Next to the *Move*, *Rotate* and *Scale* command icons are icons for the X, Y, and Z-axes. These allow you to lock the axes to better control the motion of the object. Another way to restrict motion to a single axes is to grab the X, Y or Z-axes of the object and drag. This works regardless of whether or not the axes have been locked in the Command Palette.

Translation (move) : This allows movement along the X-axis, Y or Z-axes. Whether movement is along the coordinate axes of the World Coordinate System or that of the object depends on which system is selected. Translation is relatively straight-forward. It does not depend on the object's relationship to other objects, but to whether you have selected the object or World Coordinate System.

Scale : Scale lets you change the scale **or** the size of an object, depending on the tool selected. Generally, you should use scale only as part of the animation process when necessary. Use the Object mode for scaling. However, always use the Model mode to adjust the size of an object when you are constructing and refining it, or when making one object proportional to another. Otherwise you may run into serious distortion problems when later rotating a Child of a Parent whose system has been distorted by scaling with the Object mode.

Use the Model mode when constructing objects. Use the Object mode only when animating.

Scaling with the Object mode changes the scale of the objects' axes. Since the Child objects use this axis for a reference, their geometry will become distorted when rotated in this new reference system. For more on this, read the sections in the CINEMA 4D Help files on the Object mode and the Model mode a couple of times.

Rotation : Rotation can get complicated because of the ways CINEMA 4D can handle the mathematics and the way the values for rotation are displayed in the Coordinates Manager. There are a number of rotation systems available in the "Order" parameter on the "Coordinates" panel within the Attributes Manager.

The first is the HPB system, which we look at in a moment, however the second is the **XYZ** system, this method is straight-forward and easy to visualize. If you rotate around the X-axis with the other axes locked, or drag the X-axis handle, the object will rotate around its X-axis, regardless of its (or its parent's) current orientation. The same is true of the other two axes. It all seems so easy.

Unfortunately, the XYZ system is not suited for making internal calculations involving rotation, especially during animation (see the "Help" files for a technical discussion if you are interested, especially the sections on the World System and on the Rotation Tool). So, although you can rotate the X, Y and Z-axes interactively in the editor, the Coordinates Manager (and the Coord. Tab of the Attribute Manager) will always display the values used by the internal system.

The default system used within CINEMA 4D is called HPB. This stands for Heading, Pitch and Bank (these also use the X, Y, Z keys when locking axes for use with the rotation tool). Imagine an airplane whose compass (when viewed from above) is pointing at some angle from North. This is called its Heading. The angle at which it points up (as in taking off) or down (as when landing) when viewed from the side is called its Pitch. Its Bank is how much it is rotated about its front-back axis (sometimes called "roll"). A plane typically banks (lowering one wing) while changing its heading (making a turn).

Navigation Tutorial

This tutorial follows the Navigation handout which describes the movement of objects and cameras in the CINEMA 4D viewport windows.

1. Open the file

Open the CINEMA 4D file, "Navigation_Tut.c4d".

2. Managers and windows

You have already worked with managers and windows in the first tutorial. This is a review and we will go into a bit more detail. Study the "*Overview*" diagram to see the screen location and appearance of the principal managers. Find each of these on the screen and look through some of the commands in the menu bars. Notice that each window has its own set of commands in its menu bar. Also notice that some items within a menu item have a small triangle to the right of their name. Clicking and holding the cursor over these items will expand the menu to reveal a sub-menu with additional offerings. To select one of these, simply drag the mouse over to the sub-menu and move down the items until you find the item you want. Then release the mouse to choose whatever operation or command is highlighted.

Some windows, such as the Coordinates Manager, do not have menus, even though they have text along the top. In time you will get to know which managers to use and where to find the commands you need.

Each manager window also has a shaded square icon in its upper left corner. This allows the windows to be arranged and displayed flexibly. Maybe later on we will make adjustments to this format (the GUI), but for now, *do not change the layout of the GUI*. This layout has been carefully chosen to best fit our purposes, the monitor we are using and the standard setup. Also, we all want to use a similar layout so we can communicate with each other, exchange ideas and troubleshoot more easily.

The two most common free-floating windows you will use are the Material Editor and the Timeline. They are both accessed a bit differently.

To open the Material Editor, simply double-click on any material icon in the Material Manager.

If no materials exist or you wish to make a new one, make sure you are in the active Material Manager window and enter the keyboard command *Cmd+N*. This is a command used to create a new element (window, object, etc), depending on which window is active when the command is entered. Alternatively you can double-click on the Material Manager below its menu

Use *Shift+W* to close a free-floating window. However, since this is also the way to enter a capital "W" in a text box, it will not work to close the window if the cursor is currently inside a text box.

In the Material Editor, click on the text (but not the check box) for any of the channels such as Color (so no text box is active). Enter Shift+W.

Remember, always be sure the window you wish to close is active before using *Shift+W*. Other free windows must also be opened from a menu or keyboard command.

Open the Timeline with Shift-F3 (or use the non-standard Shift+E if setup on your computer). Close with Shift+W.

The non-standard *Shift+E* has been custom installed so the Timeline to be opened and closed quickly. This will help prevent you from inadvertently pressing Shift+W. There will be times, e.g. in the middle of a complex animation, where you will open and close the Timeline dozens of times. Of course at home you can just display the Timeline on your second monitor all the time!

3. Views

Objects are grouped together in space and time into a scene. This scene, the 3D world, appears on the screen in windows called viewports. From one to four viewports can be grouped into a view panel. Each viewport will show the model from a different orientation (both viewports and view panels may also be referred to as views). The arrangement of viewports in a view panel can be varied using the options in the viewport's Panel menu. Since the view is where objects are modeled and edited, it is also referred to as the "editor window" or "editor".

For some uses, like arranging lights in a scene, you may want to work in a panel where multiple viewports are shown. Much of the time though, you will want to maximize space by using only one viewport. There are three ways to navigate between different viewports: You can select views from the Cameras menu item on the view window menu bar (slowest). You could also use the icon at the upper right of the view window (slow); but the most efficient and speedy way of switching views:

Use the function keys F1 through F5: F5: F1=Perspective. F2=Top. F3=Side. F4=Front. F5= All views (these are in the same order as the panels appear in the All View mode by default). Get used to using the function keys only, do not rely on the icons. Practice switching views with the function keys. Pay attention to the orientation of the axes as you progress through the views. Practice naming which view you are in by looking only at these axes.

4. Cameras

All views are seen through a camera. The Default Camera is provided as a means to view a scene without having to create a Camera object. If the view is not associated with a camera object you have created, the scene is viewed through the Default Camera. The camera through which each viewport is seen can be set via the view menu item *Cameras*.

Be sure the perspective viewport is chosen (f1). While in this single viewport, choose Cameras / Use Camera / Default Camera from the view menu. Try selecting different types of cameras options from the Cameras item of the view menu. When done experimenting, select the Perspective camera view.

You will rarely use this feature. Unless you require a non-standard view, **you should always use the function keys to toggle between the four preset views and the All View mode**. Each time you select a camera view from the Cameras menu item, CINEMA 4D's default settings will be applied. CINEMA 4D does not remember how you last arranged your scene with that editor camera. Also, the new camera view will now be associated with a particular F-key, making it confusing to remember which F-key is associated with which camera view. So, again, **only use this feature when you need a non-standard view** such as a bottom view or a left side view.

Select Cameras / Use Camera and choose intro Cam.

If you study the *Cameras* menu item you will note that Default Camera is now no longer hi lighted and intro Cam is now hi lighted indicating that it is the camera through which the scene is viewed in this viewport.

Try to move or rotate the view.

Does not work? This is because intro Cam has been locked with a Protection Tag.

Unfold the object ...stage.. . Notice the Protection Tag a circle with a slash through it) after intro Cam.

The Protection Tag prevents you from inadvertently moving an object whose current position should remain fixed. Notice however, that the object Cam 2 is not protected.

Select "Cameras / Use Camera / *move Cam*". Now try to rotate the view. Now it responds. Undo any rotations (remember, view/camera undos are done with *Shift+Cmd+Z*). Also, try a translation followed by an undo.

If you do not perform an undo (*shift+Cmd+Z*) after these actions the changes would replace the previous values. If too many actions are completed without undoing, they will be lost. If you select "Cameras / Use Camera / Default Camera" and perform the same operations as above, the results would appear the same. In general, you should use the Default Camera for modeling and positioning elements in your scene (although you may certainly use Object Cameras as well) and an Object Camera should be created for any view from which you wish to do a final render. Object Cameras can also be used for modeling when you need a specialty view which you can then return to anytime by selecting that camera. Any time you wish to animate a camera's motion, you must use an Object Camera.

Be sure to switch back to intro Cam in the view window before proceeding with this tutorial.

A final note on cameras: Selecting a camera in the view window menu bar will make that camera the one through which the scene is viewed. Selecting a camera in the Object Manager makes that camera the active object to be moved around like any other object. If the camera selected in the Object Manager is not the one used in the active view, moving it will not affect the current view. Nevertheless you are still changing the view as it will be seen from the camera object you are moving. Sometimes it is handy to have two view panels open, one showing a view through the camera you are moving and another from a different camera (or the editor camera) to help you position the camera being moved. You will learn how to create new view panels later.

You have noticed that, when in the Object Manager, the selected object is displayed in orange text. The active camera has a small white icon to the right of the visibility buttons. You will also notice that other cameras have this icon. However it is black.

5. Navigating in the Viewport

You have already had some practice with moving the view around. You should remember how to use the 1, 2, and 3 keys for translation, scale and rotation of the view. You should also remember to undo view (camera) moves in the editor (view) window using *Shift+Cmd+Z*.

Click anywhere within the Object Manager window and enter Shift+Cmd+A. This will make sure no objects are selected.

Translation

Notice the three axes of the World Coordinate System.

Activate the Editor Camera. Practice moving the view around by holding down the 1-key and dragging using the left mouse button. Then move in and out of the view by holding down the 1-key and dragging using the right mouse button. Watch the movement of the World axes as you move the view. Undo all moves made.

You can also move in and out using the left mouse button and the 2-key but be careful not to use the right mouse button with it or you will change the camera focus.

The view in any viewport is always through a camera, whether it is the editor camera or one you have created. This means that moving the camera through which you are viewing is the same as moving the view in the active window.

Select the Camera mode and the Move Tool. Click and drag anywhere in the view window. There is no need to hold down the 1-key since the Camera mode is selected. Try pressing the Cmd-key as you drag. When done experimenting, undo any moves you have made and reselect the Model mode.

As you move the view you will see the World origin and axes move to new positions within the editor window. CINEMA 4D still considers the origin as the place where X, Y and Z all equal zero, regardless of where (or if) it appears in the view. By changing the view orientation, you are merely changing the position of the camera through which the view is currently viewed, whether it is the editor camera (as in this case) or a camera you have created.

Rotation

Make sure no objects are selected before proceeding. This can be done by clicking in the empty space below the lowest object name in the Object Manager. Or, if the Object Manager is active, enter *Shift+Cmd+A*.

Using the 3 key, play around with rotation as you did above for translation. Try rotating with the Cmd-key also pressed (there is no need to let up on the 3 key to use the Cmd-key). Undo any changes made.

As with "Translation" above, you are not changing any coordinates of the World system. You are merely changing the angle from which you view them. This, of course, means you have changed the camera's orientation. Note that, as you change the rotation, it is as if the entire world view is revolving around the origin. This means the editor (or other) camera is swinging around the origin while rotating, trying to keep looking toward the origin.

The view can also be rotated around any point on an object in the scene as well.

In the viewport click and drag anywhere the object "Cube" whilst holding down the 3 key to perform rotations as above. Again, undo any changes.

Notice how the center of rotation has changed to the point at which you clicked on the cube. Technically this is not a rotation but a revolution: The camera is actually being translated to another position in space and rotated so that the selected object can be viewed from another side. If you actually want the camera to rotate while staying at a fixed position in space, like viewing a panorama from a hilltop, you will need to rotate the camera object itself (as we will see later on).

Zooming

Although zooming may at first glance appear to have the same effect as moving (translation) the camera in and out, it is in fact different. Whereas translation in and out changes the X, Y, and/or Z position of the camera, zooming the view changes the focal length of the camera, which alters the perspective. That is, zoom changes the apparent size of distant objects compared to closer objects in the scene and can make the distant objects look closer or farther away, depending on the focal length chosen.

Hide the "Cube" object and show ScaleArray. Change the camera used in the viewport to Zoom Cam. Select the Camera mode and the Move Tool and make sure all axes are unlocked. Observe the Focus/Zoom value in the Coordinates Manager.

Since scale and size are not applicable to cameras, the center column of the Coordinates Manager is used to show the focal length when the Camera mode is selected (the fact that the value is in the X input box has no significance).

Move the view in and out using the 1 key and the right mouse button.

You should notice that the camera moves towards and away from where you clicked in the viewport. While you are dragging, pay attention to the appearance of the cylinders relative to each other. Also, look at which values are changing in the Coordinates Manager. Only the X, Y and Z values under *Position* will changing (if you happened to have a view where you were looking exactly down a particular axis, say the Z-axis in a front view, only the value for that axis would change).

Use undos to bring you back to the initial camera view. Now drag in and out using 2 key and the right mouse button.

In the Coordinates Manager you should notice that the X, Y and Z values remain constant but the *Focus* value changes. Also, you may notice changes in the appearance of the image in the editor window as compared to using the *Move Tool* . If not, we will certainly see differences in the next exercise.

Be sure to return to the initial view using undos. Zoom out (2 key and the right mouse button), so the objects move away from you, until the focus value is about 10. Now, using translation (1 key and the right mouse button), bring the view back until the front object is about the same size in the screen as it was before the zoom.

Surely you see the difference now. A focal length of 10 really exaggerates the difference between close and distant objects like a fish-eye lens in a real camera does. We will cover focal lengths in more detail later.

The Camera mode

Although any of the above movements can be performed with any tool selected by using the 1, 2 or 3 keys, the Camera mode offers you the opportunity to view (and enter) changes to the camera in the Coordinates Manager. One other difference is the ability to lock individual axes to restrict movement to only the desired directions when using the Camera mode. However, if you wish to do this, you must use the *Move*, *Scale* or *Rotate* Tools instead of the keyboard shortcuts. Another way to restrict motion will be discussed below.

6. Moving objects

Although CINEMA 4D refers to movement along the X-axis, Y and Z as "moving", we are using the more specific term translation. Objects may also be rotated and scaled.

Translation

This works just like moving views except that an object is moved with regard to the World origin. This lets you position objects relative to each other in a scene.

Hide other objects, select and show the object Trans. View through intro Cam. Select the Object mode. This will make sure you move the object, not the view. Also be sure the object coordinate system is currently selected.

You select which coordinated system is being used with the button to the right of the Z-axis lock button. When the cube is shown, the object coordinate system is being used. The World Coordinate System will display a globe instead.

Select the Move Tool (for translation) and click and drag anywhere in the editor (view) window. Do not perform any undos.

I'm sure you can see the usefulness of this function! Notice that the object has its own axes which move with it as you move the object around in the view. Notice the difference in the appearance between the objects axes and those of the World Coordinate System.

Click on the X and Y icons (X-axis Lock/Unlock, etc.) to lock those axes. Drag in the editor window.

Note how movement is constrained to the Z-axis of the object.

Now grab the arrowhead at the end of the object's Y-axis and drag.

Even though this axis is locked in the Command Palette, it can still be moved using this technique. Try moving along the other axes as well. Also note changes in the Coordinates Manager as you experiment with the position of the object. All of these dimensions are measured between the object's origin and the origin of its parent (in this case, the World Coordinate System). We have made a lot of moves from the object's original position: Instead of undos, we will return to the initial view by another means. When initially viewing the object Trans it was located at the World's origin (0, 0, 0).

In the Coordinates Manager, enter zeros in the X, Y and Z input boxes as follows. Double-click anywhere in the X input box under Position. Enter 0. Hit the Tab key (notice the value in the next box, Y-Position, is now selected and ready to be written over). Enter 0, tab and enter 0 a third time for the Z position. Hit the Return or Enter key (or click on Apply) and the translational changes will be made.

Get used to this method of tabbing between input boxes to save time.

So far, you have been translating the object using its own coordinate system. Let's move it in reference to the World Coordinate System instead.

Click on the Use World-/Object Coordinate System so that the icon displays the globe. Lock the Y and Z-axes and move the object by clicking and dragging in the editor window.

The object now moves along the X-axis of the World instead of its own X-axis. Try it along the Y and Z-axes.

Return the object to the World origin.

Scaling

This has a different effect on an object than on a camera. As mentioned in the "Navigation ..." handout, it also has different effects depending on the tools and coordinate system used.

Using the World vs. the Object Coordinate System:

Hide any visible objects. Select and show ScaleObject. View using scale Cam. Select the Model mode and return to the object coordinate system. Choose the Scale Tool and lock the Y and Z-axes. Click and drag somewhere in the viewport (not on any-axes) to change the scale of the object along the X-axis. Undo when done experimenting.

Just as with the Move Tool, you can also restrict motion using axis handles.

Click and drag on each of the three axes and observe the results. Undo any changes.

Using the Object mode **vs. the** Model mode**:**

View using tool Cam. Hide any visible objects. Unfold and select ScaleParent. Select the Object mode (we should be using the Model mode here) and the Scale Tool. Scale the object only along the X-axis by a factor of about two.

You should now be familiar with two different methods of doing this.

Notice the entire parent-child object is scaled, including the space between elements. Now for the problem with using the Object mode here.

Select Cube **and rotate it around its bank-axis (the Z-axis button).**

You probably didn't want that. When you scale an object with the Object mode, you distort its coordinate system. When Cube is rotated within that distorted system, its geometry is thereby distorted.

Undo to the initial state again. Select the Model mode instead and perform the same sequence of operations as above, scaling ScaleParent and then rotating Cube.

Notice that the cube is not distorted this time. Do not use the Object mode until you are ready to animate. It is best to always scale using the Model mode until you really get to know the program. The Object mode changes the scale of the axes, the Model mode changes the scale of the surfaces. Use the Model mode when modeling, use the Object mode when animating.

So, what if we wanted to scale the Parent object only? With proper planning we would have done that before we grouped the two objects. However, it can still be done by selecting the Parent object (in this case, ScaleParent) then scaling while pressing the 7-key. This only works with the Model mode . You can use this to independently move or rotate a Parent object without moving the Children (try it). The disadvantage is that you cannot use the Coordinates Manager to make precise adjustments using this method.

Rotation

Rotation behaves a bit differently since a different system is used. The reasons for this are addressed in the "Navigation..." handout and in relevant sections of the Help files. Here we will see what the HPB rotation system looks like and begin to get a grip on understanding how to work with it. First a look at the XYZ system and its shortcomings.

Hide ScaleParent, show Rot XYZ and view through the rot Cam camera.

Make sure to turn off the HPB system: In the Attributes Manager select the Coordinates panel and choose "xyz" from the Order parameter menu.

Now the XYZ rotation system will be utilized.

Select the Rotate Tool. Rotate around the Y-axis by dragging on the axis band (green band). Observe the Coordinates Manager.

No surprises - the Y value in the manager changes as you rotate the cube interactively.

Undo the rotation. Rotate the X-axis (red) about 30°.

Pay close attention to the next function:

While the cube is still rotated, grab the Y-axis again and rotate and observe the changes in the Coordinates Manager.

The numbers change wildly! But your cube still behaves as predicted.

Undo the last rotation (leaving the first 30° X rotation). Open the Coord. Tab of the Attribute Manager.

Instead of the Coordinates Manager, we are going to use the Attribute Manager because it lets you adjust values "on the fly" without having to hit "Enter" or "Apply" .

Click and drag the up/down arrowheads to the right of the heading text box in the Attribute Manager (labeled R . Y) up and down and observe the rotation of the cube.

At first glance this is probably not what you expected. Before reading on, drag a bit more and see if you can identify the type of rotation.

Careful examination reveals that the cube is indeed rotating around a single axis, but not its own. It is rotating with regard to the World's Y-axis (in this case its parent). This is called "heading". At first this motion appears more cumbersome to analyze and predict. However, notice in the Attribute Manager how only one value has changed. Think of how much easier this will make it if you have to animate a rotation of a specific number of degrees.

But what if I have to rotate the object around its own axis and that axis must be tilted in the World view?

Return Rot XYZ to 0, 0, 0 using the Attribute Manager and hide it. Unhide and unfold Rot XYZ Parent. Select Rot XYZ 2 and, in the Coord. Tab of the Attribute Manager, rotate its heading by dragging on the arrowheads.

It seems you can have your cake (have only one value change) and eat it too (get motion around a tilted axis). Why did it work this time?

Note the direction of the Y-axis. Now select the cube's parent Rot XYZ Parent and note its Y-axis.

The cube was actually turning around its parent's axis. It so happened that its own axis had the same orientation. This is, of course, no coincidence. Look at the icon to the right of the Parent's name: This in the sign of a "null". Null Object Objects have no geometry of their own but can be used to group other objects and, most importantly, to set up complex rotations for animation. Imagine the earth rotating around its own tilted axis as it revolves around the sun.

A closer look at the HPB system

Hide currently displayed objects. Select and show RotParent. View through rot Cam. Select the Model mode and the Rotation tool. In the Coordinates tab of the Attributes Manager re-select the HPB System.

You now see the coordinate axes of the World system with the long thin lines labeled X, Y, and Z. You can also see the object RotParent with its object axes emphasized by arrow-shaped objects. The object is named RotParent because of how it will be used later. For now, remember it is still a Child of the World.

To get a good feel for how this works, make all rotations by grabbing the axis bands and dragging. Before continuing, study the axis lock buttons in the tool palette. Forget about the X, Y and Z-axes. They become, in order, the H, P, B axes of rotation.

Heading is green, meaning you drag the green band to rotate the heading. If, for example, the Parent object is the World this will rotate the object around the World's Y-axis. This means the object rotates on the World's Z-Y plane. Another way of describing "heading" in this case is that changes in "heading" will rotate the object in a circle, around the Parent's Y-axis. Keep an eye on which values are changing in the Coordinates Manager as you do the following exercises.

Rotate the object RotParent by changing its heading (click and drag on the arrows next to the H value in the Attributes Manager). Once you have a feel for heading, undo any rotations.

Rotation occurs around the Y-axis of the Parent (the World in this case). Now let's try pitch.

Rotate the object by changing its pitch (click and drag on the arrows next to the P value in the Attributes Manager).

This may appear a little less obvious. Remember the airplane analogy: Pitch is when the plane (pointing along the Z-axis) noses down (or up). Let's try bank.

Undo any rotations. Rotate the object, changing its bank, by grabbing and pulling on the blue band.

The term bank naturally makes more sense when used in reference to airplanes than to humanoid figures. All of this was relatively easy to visualize because of the human-like figure. That is because the figure is lined up with the World axis system, which is being used as the reference for rotation. Its orientation really has nothing to do with the actual object we are concerned with, RotParent. Let's complicate things a bit.

Hide the Figure object. Now try the heading, pitch and bank rotations (each followed by undos).

It probably became a little harder to visualize. To see what's happening with pitch, you must put yourself in the "airplane" looking forward (along the Z-axis).

Undo any rotations, make sure RotParent **is still selected. Rotate the view until you are looking straight down the Z-axis of the** RotParent **object (the "airplane") (the -axis is the front-back axis of an object).**

OK, so this is the axis which is going to "nose down". However, relative to what? Relative to the Parent object (the World in this case). If you were nosing down in a plane, you would probably reference your dive to the horizon (the X Negative Z plane).

Grab the pitch handle and rotate RotParent**.**

You should see the Z-axis (the axis of our "plane") nose down (or up) into the X Negative Z plane of the World. As we adjust the pitch, the Z-axis will rotate in a plane which is always vertical within the Parent system. That is, the plane in which the Z-axis rotates is perpendicular to the World's X Negative Z plane and passes through the World's vertical (y) axis.

Undo rotations of the object and of the view. Try some more pitch rotation from this view and get familiar with the look.

Again, you should note how the object pivots such that its Z-axis dips or points upward to the horizontal plane (the World's X Negative Z plane).

Undo. Rotate the object by changing its bank. Grab and pull on the blue band.

Now the object rotates around the Z-axis in a way you probably expected. Bank always behaves in a predictable manner (actually, so does heading and pitch. It just takes a little longer to get the hang of predicting!)

OK, here is where it gets sticky! If you have problems with this, and you probably will, it will not affect your ability to learn and create good projects with CINEMA 4D.

What if the Parent object is not the World?

Let's see what happens in the case where a Parent object is not oriented the same way as the World. For this example, the object RotParent will be used as the Parent.

Undo any rotations. Unfold RotParent, select and show RotChild. Rotate RotChild about the heading.

Remember, "heading" refers to the Parent's vertical axis. Let's make it easier to see.

Rotate the view until you are looking down the Y-axis of the Parent RotParent. Now rotate the heading of RotChild in the Attributes Manager.

The rotation should obviously be in circles around the axis you are looking down (and in the plane of your monitor, which is the X Negative Z plane of the Parent). Now for a look at that annoying pitch.

Undo any rotations of both the object and the view. Rotate the pitch axis of RotChild.

You would need pretty good spatial comprehension skills to see this the first time around. Actually, the same thing is happening as before: It is just a bit harder to see because the "horizontal" (the X Negative Z) plane of the Parent does not have a grid to make it obvious, as the World does. Here we will give the

Parent an X Negative Z plane that you can use as a reference.

Show the object XZ-plane (RP). Rotate the view until you are looking down the Z-axis of RotChild and rotate its pitch.

Hopefully you will see the Child's Z-axis dive into, or out of, the Parent's X Negative Z plane. We will now make the relationship even easier.

Undo any view or object rotations. Unfold RotChild and show XY-plane (RC).

This plane defines the X-Y plane of the Child. If the Child object were our airplane, this is like the plane going up and down and side to side as you are facing forward in the plane.

Undo any view or object rotations. Rotate the pitch of RotChild.

Now you should really see that Z-axis dive into the X Negative Z plane of the Parent.

Undo any view or object rotations. Rotate the bank of RotChild.

Thank goodness, bank still behaves as predicted.

A little HPB practice

Let's try a bit of practice before leaving the topic of rotation.

Undo any view or object rotations. Rotate the heading of RotChild.

Study the motion while you rotate. Hopefully, what "heading" does is becoming fairly obvious.

Leave the heading at a new position without undos. Rotate the pitch of RotChild.

Continue rotating the different axes alternately and see if you can predict how the motion will appear before each new move.

When you are done, close the file without saving.

To remind yourself of why we use this system instead of X, Y and Z around the object axes, pay attention to the values in the input boxes. They only change one at a time as you move heading, pitch or bank. This means you can precisely set them from the input boxes and from tweaking FCurves as we will learn later. If you were to use an X, Y, Z system, two or three of the values would change immediately with no hope of manipulating rotation any other way but by eye. Although this will appear quite satisfactory for many purposes, sooner or later you will want or need to set rotation values numerically, especially when animating. If this rotation stuff seems a bit daunting, do not worry, you will pick up what you need with practice.

Introduction to Modeling

Overview

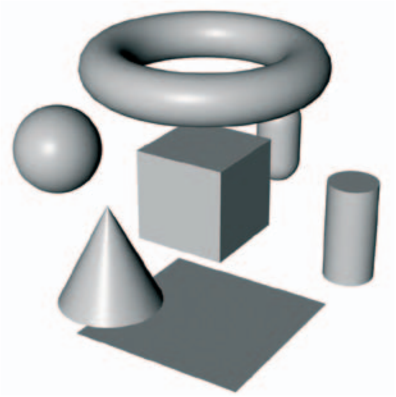
Modeling is the process of creating an object which can be textured, illuminated, animated and rendered to look like a real or imaginary object. While you are in the process of modeling, the object may be referred to as an "object" or a "model". CINEMA 4D generally uses the term "object". Once modeling has been completed, CINEMA 4D always refers to it as an object. Objects are accessed via the Object Manager. Here they can be selected, hidden or unhidden, unfolded to show sub-objects (children) or have attributes assigned to them by way of tags. The tools for creating objects are found in the main menu bar. The most important or most commonly used tools are also found as icons in Command Palettes along with important navigation tools. Some tools also have keyboard shortcuts to speed up the modeling process.

Most complex objects have humble beginnings. The **object primitives** we will begin with are quite often used as a first step in creating a model. CINEMA 4D also has many tools for modifying these primitives and the shapes which can de built from them. In a later tutorial, for example, you will begin with a cube and end up with a plastic milk bottle.

Another commonly used method of starting the modeling process is creating curved lines called Splines. These Splines can be manipulated in various ways by tools called **Generators** to create 3D surfaces. Later, you will model a table lamp by creating a Spline outlining the profile of the lamp which can be swept in a circle ("lathed") to create the base of the lamp. Although the above methods are powerful and can even be the only techniques you need to model certain objects, additional methods will be presented later which offer very fine control over the geometry of an object.

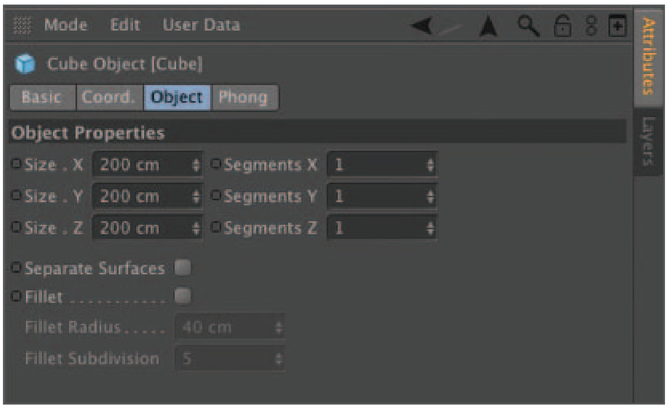
Primitives

Several types of primitives are available. They can be found in the main menu Create / Object or in the top Command Palette. Each of these object primitives has a number of different values associated with it. These values can be adjusted to modify the shape of the primitive. These adjustable aspects of a primitive's shape are called parameters. The primitives in CINEMA 4D are all parametric objects, i.e. their shapes can be modified by user adjustable parameters.



Parameters

A primitive's parameters can be modified by two principal means. The first is through the Attribute Manager which shows the object's controls and parameters whenever the object is selected. Here you can directly enter precise values into input boxes just as you have previously done in the Coordinates Manager. Some parameters can only be accessed this way.



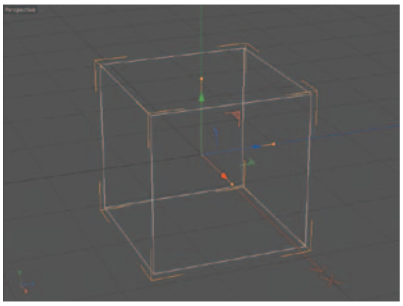
Cube Object [Cube]

Object Properties

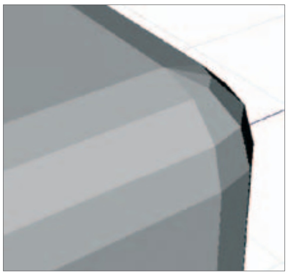
○ Size . X	200 cm	○ Segments X	1
○ Size . Y	200 cm	○ Segments Y	1
○ Size . Z	200 cm	○ Segments Z	1

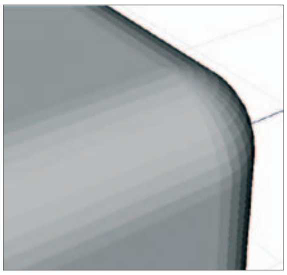
○ Separate Surfaces □
○ Fillet . . . . □
Fillet Radius . . . . . 40 cm
Fillet Subdivision 5

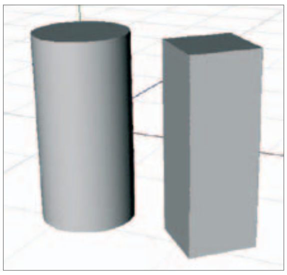
The second method of manipulating parameters is to do so interactively in the editor window. Each primitive displays orange handles with little orange squares at the end which can be grabbed and dragged. You may also use the HUD (heads-Up Display) to make adjustments. Some parameters, such as size, are also adjustable in the Coordinates Manager.



Segments





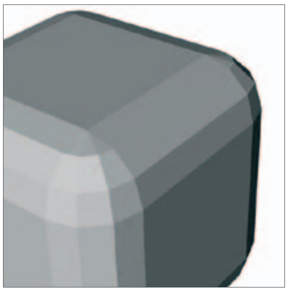


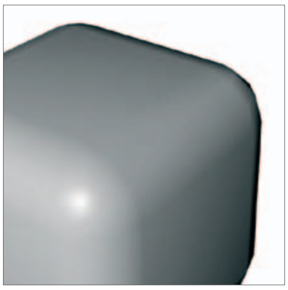
One of the parameters you will see in the Attribute Manager for almost every primitive is *Segments*: Some primitives will have more than one input box for this. This is quite important for two reasons. The number of segments determines how the object appears on the screen. Objects with fewer segments appear faceted or block-like. More segmentation gives a smoother, more curved look but requires more memory and processing time. The top cube at the right has only three segments that define its filleted (rounded) edge. The bottom cube uses ten segments. If you were to create a cylinder it would appear rounded because the default number of rotation segments is set to a relatively high value of 36. This means the cylinder's shape is not defined by a perfect circle but by 36 short straight lines joined end-to-end to approximate a circle. If you change this value to 4, the object no longer appears to be a cylinder but a long, rectangular solid (here, our circle is made from only four segments and therefore appears to be a square).

The other reason segmentation is important is that it determines the number and placement of polygons and points in the object. This becomes important when modeling complex objects built from these primitives. More on this when we discuss Polygon Modeling later.

Choosing the proper number of segments is important. You must make a compromise between a smooth look (if that is required) and memory/time considerations. Various factors such as how close the object will be to the camera, how long it will be on the screen, final modeling techniques and how the object will be textured all play a role in how many segments to use.

Smoothing





Although we will talk more about tags in the section devoted to the Object Manager, one is special enough to be included here because of its relationship to segments. The **Phong Tag** is used to give objects a smooth appearance. A primitive without a Phong Tag will appear faceted. The number of facets will correspond to the number of segments you have set in the Attribute Manager. Using smoothing causes CINEMA 4D to calculate a smooth transition in texture from one segment to the next, making the surface appear smooth (within limits). This is quite useful for most models as it allows use of a much lower polygon count, thereby saving memory and rendering time but still giving an acceptably smooth surface. However, smoothing works by applying the calculations to the texture, not the object itself. This means the silhouette of a smoothed object will still show faceting. The two cubes pictured at the right are the same with the exception of a smoothing tag. Notice however, that the silhouetted edge of the lower cube is still faceted. The smoothing only appears on the surface of the cube.

Sometimes you may want hard edges on some parts of an otherwise smooth surface. This is what the Phong Tag's Angle Limit parameters are for. Any surface facets (polygons) that are at an angle greater than the angle limit will not be smoothed.

Displaying objects

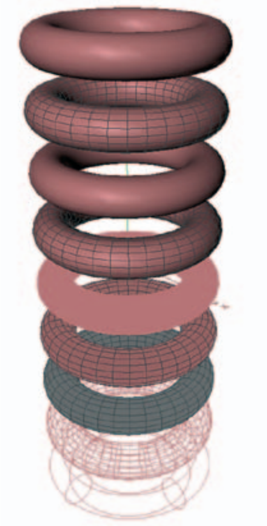
Objects may be displayed in several ways. Furthermore, different objects may be displayed in a different manner in the same scene. Some modes show the best detail for activities like modeling. Others are better suited for quick animation previews in the editor window to better judge the movement of objects. The display modes are accessible from the Display menu. As you go down this list, the principal distinguishing characteristic is the increase in the redraw speed, however at the expense of detail. Here are the possible choices:

Gouraud Shading : The "full service" mode, shows textures and the effects of all lights. This is the mode you will use most often. *(top torus)*

Gouraud Shading (lines) : As above with the addition of lines showing the underlying polygon structure. *(2nd torus)*

Quick Shading : Shows textures but uses the single auto light instead of your scene's light objects. (3rd torus)

Quick Shading (lines): As above with the addition of lines showing the underlying polygon structure. (4th torus)



Constant Shading : Objects are shown with flat, uniform lighting.

Constant Shading (lines) : Objects are shown with flat, uniform lighting along with the underlying structure.

Hidden Line : As above except lines hidden behind front surfaces are not shown. *(7th torus)*

Lines : Shows only the underlying structural framework, no surfaces are shown.

The following are used in conjunction with the modes above:

Wireframe : Shows polygonal lines (if used with a mode above displaying lines). *(8th torus (used with Lines mode))*

Isoparms : Even faster to draw. Shows minimal wire outline of certain types of objects. These objects allow the number of isoparms to be defined. *(bottom torus)*

Box : Very fast. Uses a simple box in place of the object.1

Skeleton: Extremely fast. Only useful to show animated relationships of hierarchically grouped objects. Shows lines connecting objects.

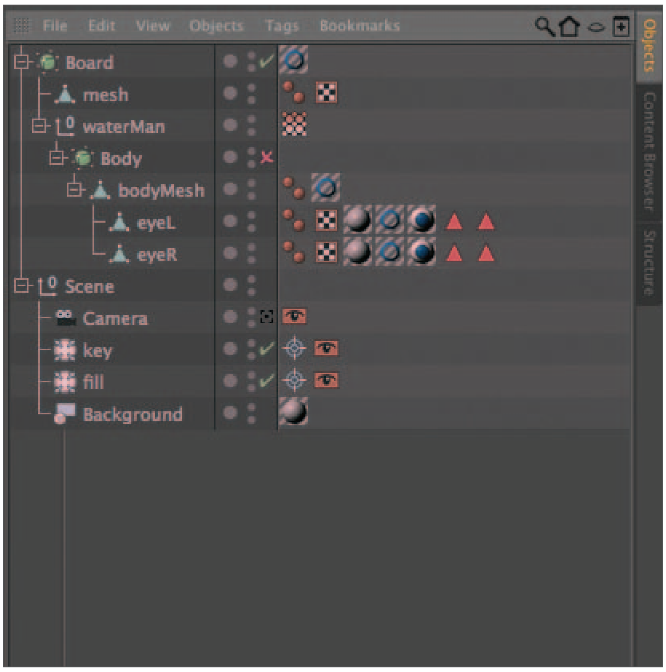
You can have additional control by mixing render types on the screen. From the options menu, "Configure" brings up the viewpoint settings in the Attribute Manager which allows you to set the mode for the active object to a different one than the rest of the scene. Also, in the Object Manager, under "Tags / CINEMA 4D Tags / Display", you can set the display mode for a particular object that will override that of your selection in the editor window (unless the check box Display Tags has been unchecked in the Configure settings). This is useful when working on modeling an object where you need to keep a number of other complex objects visible for proper fitting but the redraw times are too high when all are displayed in Gouraud Shading.

Level of detail

Under the view menu item Options / *Level of Detail*, you can select the amount of detail (*low, Medium, high*) for objects to be displayed in the editor. This does not affect how objects are rendered either in the editor window or in the Picture Viewer window. It only affects the editor while you are manipulating and arranging objects. This setting's purpose is to reduce redraw times while you work in the editor.

The other choice in Options / Level of Detail is Render LOD (level of detail). This also has no effect on final rendering, which is always done in the Picture Viewer but does affect rendering in the editor window, which is where you frequently render to get an idea on the outcome of certain operations. Checking (activating) this item means that the objects will be rendered at the highest settings you have set for them in the Object Manager and in the *Edit Render Settings* window.

The Object Manager



You have already done some work in the Object Manager. Now we will fill in a few details and offer a preview of some attributes that will be essential for complex animation later on. The first column in the Object Manager contains a list of object names. Names may be changed at any time by entering a new name in the input box which pops up when you double-click on a name. Important objects, those that will be referenced (have links to them) by other objects or will be animated individually, they should be given unique names. Names can be changed and CINEMA 4D will update any references automatically. To the right of the object's name is an icon showing the object's type. Clicking either on the name or the icon of an object will select that object and load its attributes into the Attribute Manager where they can be controlled through input boxes and check boxes.

The three buttons in the center column control whether an object is assigned to a **Layer** and whether or not the object will be displayed. The larger of the three is the layer assignment button. Clicking on it will trigger a dialog asking you if you want to assign it to an existing layer or create a new one. To the right are the two display buttons. The top button controls display in the editor. The bottom button controls display during rendering (both in the editor and to the Picture Viewer).

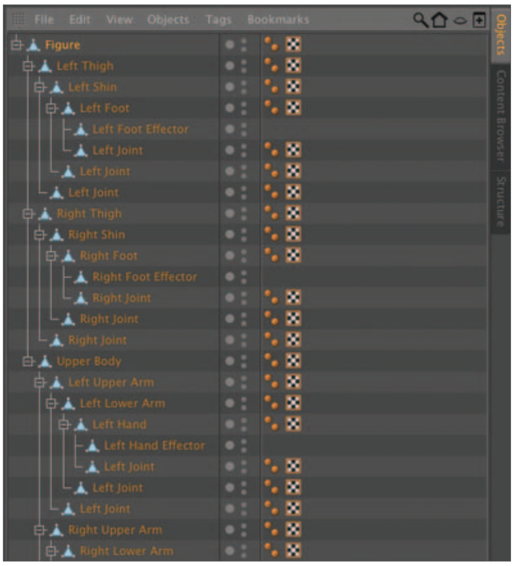
Gray, the standard mode, represents the "on" (visible) mode unless a Parent object is "off" (hidden). Red is "off" regardless of the Parent's state. Green is "on" regardless of the Parent's state. The check mark/X in the second column turns certain tools such as deformers on and off.

Object tags

The third column displays an object's tags. Tags give attributes to objects and, when clicked, load their attributes into the Attribute Manager allowing adjustment of their parameters. Once you have defined an object's basic shape, tags allow the object to take on personality such as smoothing, texturing, method of texturing, limits to the motion of some objects, whether the object should receive or cast shadows, and much more. Most of these tags will be discussed later since they involve other aspects of production, such as texturing.

We have already mentioned two tags, *Phong* and *Display* . Another, which you might soon find a use for is the *Protection* Tag. This tag locks the motion of an object, preventing you from accidentally selecting and moving it.

Object hierarchies



A hierarchy of objects is the relationship in which some objects (**children**) are linked to others (**parents**) in such a way that they inherit certain properties from the Parents, especially motion. This is essential for all but the most basic of animations. In fact, many objects are not single constructs but are built from multiple objects organized in a hierarchy. These hierarchies are reflected in the positions of objects in the manager.

An object at the top of any hierarchy will have a small box to the left of its name. If the box contains a plus, the hierarchy is folded and the sub-objects (children) are not visible. A single click in the box will unfold the hierarchy to reveal the next level. All objects under the Parent will be indented and displayed under the Parent.

Some of the Children will also be parents to their own Children. This can be recognized by "+" boxes to their left. As you unfold these, the indentation will continue further to the right. Some complex objects may have many "generations" of Children to yield the desired effects when animated. You should unfold and fold these hierarchies as needed to keep the manager uncluttered, while allowing access to the particular object you need.

Although hierarchies are important to help organize your workspace, they are indispensable for setting up animations. As an example, consider a car. You might wish to use the body object as the Parent. Some of the Children would include the tires. Wherever the body goes, the tires will go as well. However, as they are Children, you can rotate the tires without rotating the body (the motion is inherited down the hierarchy, not up). If you wanted hubcaps on the tires, you would make them Children of the tires to which each belongs. As a Child, the hubcap will now rotate with the tire.

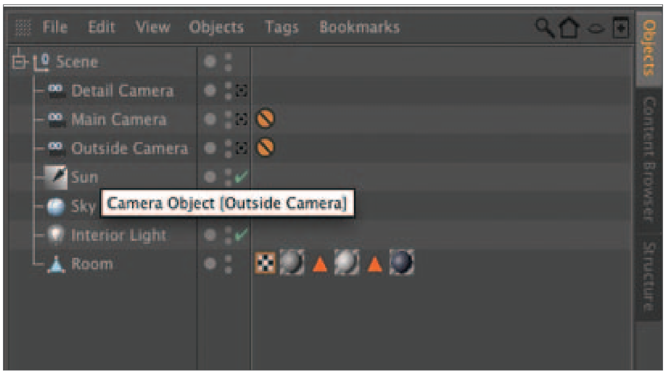
Arranging objects

It is easy to arrange objects in the manager, just drag & drop. Click on the name of the object and drag up or down to the desired location. If a left arrow appears at the position of the cursor when you release the mouse button, the object will be placed at the same level (in the hierarchy) as the closest object. If a down arrow appears, the object will become a Child of the object beneath the cursor.

You may also copy objects via drag & drop. Hold down the *Cmd*-key while dragging and move the copy to the desired location. When done this way, the new object will have the same name as the old. Change its name if you animate it or refer to it from another object. You may also cut and paste to duplicate objects.

Grouping objects

One simple arrangement of objects, not exactly a hierarchy, is the group. A group is made up of one "parent" object with many sub-objects ("children") at the same level. A group is only one level deep. This is commonly used when you want to move several objects as a group, assign specific attributes to a large group (allows for easier changing later) or just to keep the Object Manager uncluttered. This last application is actually quite important. Complex scenes with hundreds or even thousands of individual objects may only show five or six objects when all are folded. Imagine 40 rocks at the bottom of a stream. Do you want to scroll past all these to make your way from the water to the fish in the Object Manager?



Grouped objects generally make use of a *Null Object* . A Null Object contains no renderable information. A Null Object is used in this case to organize other objects (they have other very important uses in animation to be discussed later). So grouped objects often have one of these Null Object Objects as their Parent. You can make a Null Object from the main menu: "Create / Object / Null". The objects to be grouped may then be dragged into this Null Object and the Null Object can be named to reflect the contents of the group (i.e., "BottomRocks") The Null Object may be animated like any other object. In this manner, many objects may be moved together as a group. Another way to group objects is to select multiple objects and press the *alt+G* or *Option+G* key.

Modeling Intro Tutorial

This tutorial follows the "*Introduction to Modeling*" handout that describes how to use primitives and the Object Manager. Unlike the previous tutorials, this one will begin with an empty scene and you will create the objects yourself.

1. Primitives intro

Primitives are the most basic of 3D objects. They do not consist of points and polygons (although these are calculated when a primitive is rendered) but are constructed from mathematical formulae. You could write these formulae by yourself if you knew higher level math. However, CINEMA 4D lets you choose from a number of pre-constructed shapes. All you have to do is feed data into the formulae so CINEMA 4D knows the exact specifications for a given shape. The different values you can specify for a given shape are known as **parameters**. This is why primitives are also known as parametric objects.

Open CINEMA 4D. Make sure you are in the perspective view (with just a single viewport in the view panel) and, using the Camera mode, set the Focus to "60" in the Coordinates Manager. Switch back to the Model mode. Make sure the Display menu option is set to Gouraud Shading. Save the file in the "Intro Tutorials" folder as "Modeling_Intro". Save frequently, especially at the end of each day's session so you will not have to start over again the next day.

This is generally a good view with which to begin any modeling session. Most of your work will be done in the perspective view. Other views will be used as needed for construction details and for placing objects in the scene relative to one another.

Create a Cube Object from the main menu, Create / Object / Cube. Your cube will appear in the Object Manager with the generic name Cube at the default size. Note that Cube will appear in orange text, indicating that it is the selected object. Press the delete key.

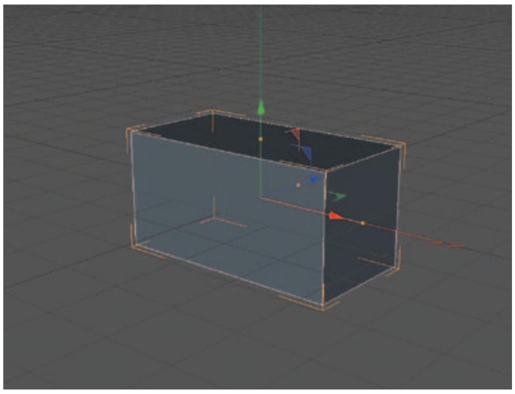
The "delete" key will delete the currently selected object (unless you are using the point or Polygon modes with the viewport active). You could also have performed an undo to eliminate the cube since creating it was the last action performed.

Click once on the Cube icon in the top toolbar.

Same results as before. Although you will most likely prefer and become accustomed to using the toolbar (command Palette), you should know the location of the more important commands and tools in the pull-down menus. One reason for this is that the menus group these commands and tools into categories which help you recognize the organization and flow in such a complex program.

Remember to use the Model mode for modeling. Establish good work habits early. First, you will change the dimension of the cube "interactively".

Drag the orange handle aligned with the X-axis until the cube is about twice its original length.



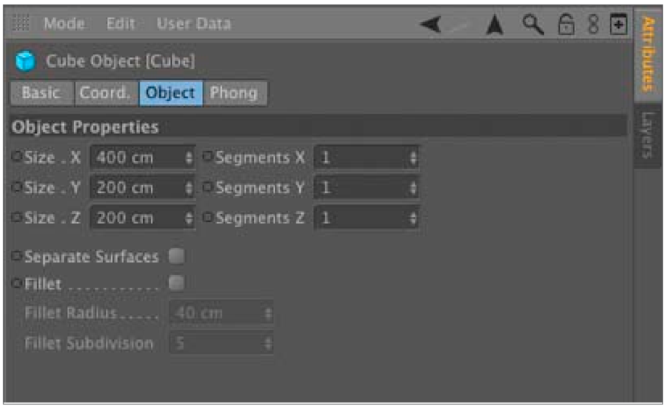
Pretty simple. This method works best if you are trying to adjust the relative size of one object to another as they appear in the editor and exact dimensions are not important. If you need more precision than this eyeball method, enter exact values in the Attribute Manager .

With the Cube Object selected, make sure the Object Tab is displayed in the Attribute Manager.

In the Attribute Manager, "Tabs" refers to the row of boxes under the object name which show or hide different groups of parameters. Shift+click shows a group, Cntrl+click hides a group and right-click toggles between states.

Set the Width (X) to 400. Press Return (or Enter).

(You do not have to hit *Return* to cause the newly entered value to take effect. Any subsequent action such as tabbing to the next input box or clicking anywhere else has the same effect as *Return*). Using the Attribute Manager will give you precise results for such objects as machined parts or a room with specific dimensions. Remember, you can click and scroll on the small button at the right side of the input boxes to also change the value of a parameter.



Cube Object [Cube]

Basic | Coord. | Object | Phong

Object Properties

Size . X 400 cm | Segments X 1
Size . Y 200 cm | Segments Y 1
Size . Z 200 cm | Segments Z 1

Separate Surfaces □
Fillet . . . . □
Fillet Radius . . . . . 40 cm
Fillet Subdivision 5

Let's adjust another parameter of cubes.

In the Object Tab of the Attribute Manager, enable Fillet by clicking on its check box.

What if you want less rounding?

In the Attribute Manager, change the Fillet Radius to 10 and either hit Tab, Return or click in an empty area of the Attribute Manager.

Smaller value, less fillet (rounding). The subtle use of this is common. Most real-world objects do not have the super sharp edges of the default primitive objects.

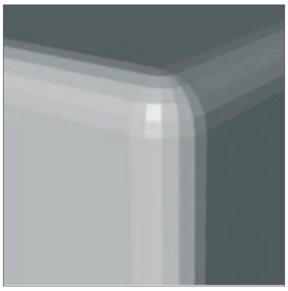
A note on selecting objects: Objects may also be made active (selected) by clicking on them in the editor window. This method of selection is best done with simple scenes or isolated objects. When there are many objects in close proximity to one another it becomes difficult to select the one you want and it is usually easier to select objects via the Object Manager.

2. Segments

Before we go on to survey other primitives, let's take a look at segmentation. Later we will find that this directly affects the number of polygons in an object.

For now we will look at how it affects the smoothness (faceting) of a surface. When a primitive is created, a *Phong Tag* is automatically applied, which smooths the surface but not the silhouette. We will get rid of this tag to better see the segments (subdivision) created by the fillet operation.

Using Cmd+1 with the mouse, move the camera closer to the corner of the cube nearest the camera until only about half the cube can be seen in the view. Also, drag the view while pressing the 1-key, without Cmd if necessary, to center the corner in the view. Now delete the Cube's Phong Tag.



You should be able to see that the edge rounding is no longer smooth but consists of long facets (segments) parallel to the edges. Count them.

In the Attribute Manager, note the number in the Fillet Subdivision input box (does it match your count?). Now change the value to 1.

A carpenter would call this a chamfered edge. For some applications, one segment will give you the look you need.

Set the number of fillet segments to 50.

This certainly looks quite smooth. The edge is broken into so many facets that it appears to be a continuous curve. However, there is a price to pay.

Drag the view around rapidly in circles in the window (holding down the *1*-key and dragging).

Notice the hesitation in the movement. Lots of calculations are being made every fraction of a second (if you are working on a very fast computer, you may have to increase the value to 200 or more to produce a noticeable lag).

Return the view to center (shift+Cmd+Z). Change the fillet segments back to 1 (fastest way is with an Undo). Again, drag the view around.

You should notice a much faster response. The fewer number of segments, fewer number of calculations that have to be made.

Change the Fillet Segments to 150 and drag the view around.

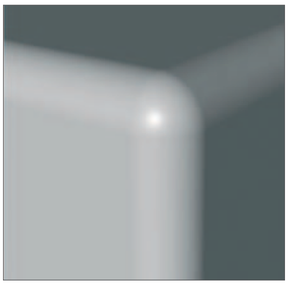
Depending on the speed of your computer and settings in the preferences, one of two things will have happened. The movement may be very slow and the image hesitates and jumps. CINEMA 4D may lower the display mode to wireframe in an attempt to keep up with the movement.

Even though you are working with a parametric object (no polygons to manipulate), CINEMA 4D still translates the parametric information into polygons to produce an image on the screen. More segments = more polygons = more memory + more time to calculate and render.

This is a very important point. As you add more objects (with more polygons) to a scene the problem becomes progressively worse. It is easy to spend a lot of time creating a scene with too many polygons to render.

Reposition the view so the front corner occupies most of the view, as at the beginning of this section. Render the scene in the editor (render Active View).

Note the smoothness of the object and, as shown in the status bar at the bottom of the screen, the time it takes to render.



Change the Fillet Segments to 10 and render.

This should render faster and be almost as smooth.

Change the Fillet Segments to 5 and render.

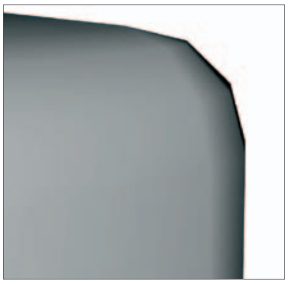
This should render even faster but the faceting begins to show. Leave the fillet settings at 5.

Move the view away (back up the camera) until the Cube Object is a bit less than half the width of the screen. Render.

It should now look a bit smoother, probably acceptable for many uses if it is not visible for too long or if we do not move in much closer. By now you should see that using the fewest number of segments (polygons) is in the best interests of speed and memory conservation. However, there is another way to smooth surfaces without increasing the polygon count.

3. Smoothing

Smoothing is a "cheat". It does not really make the mathematical surface smoother. It affects how the material is applied to the surface. It turns out that this is a much faster, less memory-intensive way of producing a smooth looking surface. However, smoothing has its own limitations which we will see later. For now, let's see what smoothing can do.



Return the view so the front corner occupies most of the view as at the beginning of the previous section. Set the Fillet Segments to 3 Render.

Note the render time.

Very faceted again. This time, however, little memory is used and rendering was faster (the object can also be moved rapidly in the view).

Add a Phong Tag to the Cube (be sure it is selected) using Tags / CINEMA 4D Tags / Phong from the Object Manager menu. Render (figure at the right) and note the time. Drag the view around.

This seems to create a similar effect as a high facet count with faster results. Most of your objects will use the Phong Tag to avoid higher polygon counts. However, like all good things, it has its price.

Study the image at the right. It is a rendered version of the same cube you also created, shown in profile at the corner. The Phong Tag smooths the texture on the surface but it does not affect the actual geometry of the cube, as can be seen by the facetted silhouette. We will have to increase the subdivision to improve the smoothness of the corner. The default value of 5 will work fine for most cases. A close-up like seen at the right may require 10 subdivisions if we get a good long look at it. Later, we will learn about a powerful tool (the HyperNURBS deformer) that will make the edge look quite smooth when viewed up close with only 5 subdivisions.

4. Other primitives

Now that you have an idea of what a primitive is and how to adjust its parameters we will look at a sampling of other primitives.

Sphere

Return the editor camera to its default view. From the view menu bar, select View / Frame Default. Delete the object Cube and create a sphere.

You can do this from the main menu, Create / Object / Sphere or from the toolbar by holding down on the cube icon until it expands to reveal the other primitives, then dragging the cursor over the sphere icon and releasing.

Any of the tools in the toolbar that have a small triangle in the lower right will expand in this manner to reveal other, related choices.

Move in on the view to enlarge the object's appearance (cmd+1 drag). Change the display mode to Gouraud Shading (lines).

Now you can see the underlying geometry which CINEMA 4D will use to render the object.

Delete the Phong Tag. Note the parameter Type in the Attribute Manager.

The default sphere is *Standard*. It is segmented like the earth is in latitudinal and longitudinal lines. Notice the *Render Perfect* box is enabled.

Render.

You can change the sphere is parameters as you please - it will still render as a perfect sphere if this box is enabled. This is unique to the Sphere Primitive.

Uncheck Render Perfect and render again.

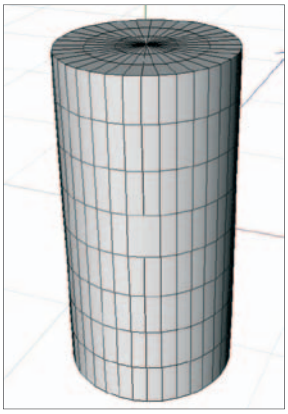
If you were to use a smoothing tag with the *Render Perfect* box unchecked it would behave the same as the cube and other primitives. The surface would appear smooth but the silhouette would reflect the number of segments chosen in the Attribute Manager. Render Perfect with a low segment count is fast and the way to go when your object is to remain a perfect sphere.

The other sphere types, with the exception of the Hexahedron, are of little interest.

In the Attribute Manager, change the Type of Sphere to Hexahedron.

The Hexahedron is useful because it is composed entirely of quadrangles (four-sided polygons) with no triangles. This is more desirable if the sphere is to be converted to a polygonal object and modeled further, e.g. into a head.

Cylinder

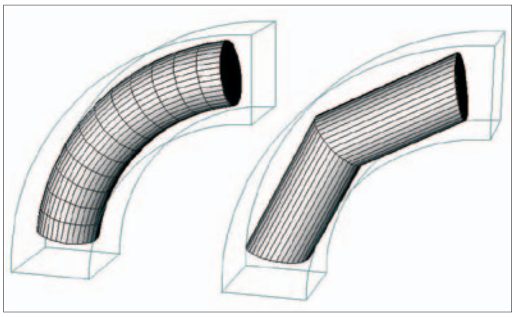


Delete the sphere and create a Cylinder Object. Delete the *Phong* Tag. Rotate the view somewhat so you can see the top of the cylinder as well as its side. Make sure both the Object and Caps Tabs are visible in the Attribute Manager.

The number of lines ("wires") displayed is a direct result of the segmentation parameters you set. Compare the values for the segment parameters in the dialog to what you see in the wireframe object. Especially notice the Cap Segments. Many of the primitives have caps, as do the Generator objects we will discuss later.

Change the *Height Segments* to 4, the *Rotation Segments* to 6 and the *Cap Segments* to 2.

Now you should better see how the segment parameters are related to the model. By now you can probably relate the number of rotation segments to the smoothness in the circumference of the cylinder, especially in its silhouette when looking from above. However, what about the height segments? It seems that one is as good as eight. This would be true if this were the final form of your object. However, if you intend to use deformers to change shape or animate the object, the segments will show. For instance, bending the model with only two height segments will result in the appearance of a v-shape instead of a smooth curve (see the figure above). We will return to this under deformers. As always, the number of segments determines the number of polygons if the primitive is converted into an editable (polygon) object for detailed modeling.



Try checking the Fillet option. Try Slice as well (in the Slice Tab).

Cone

Some of these objects are more versatile than they first appear. We will look at the cone as an example.

Delete the cylinder and create a Cone Object. In the Attribute Manager, change the Rotation Segments to 4. Delete the Phong Tag.

Voila, a pyramid. A Pyramid Primitive is also available but using a cone is more versatile. A cone can be assigned any number of sides. It can also be truncated (cut off at the top).

Change the top radius to 20. Experiment with the parameters interactively.

You can even create a cube. Of course you could not fillet the edges like you could a Cube Object (also, the cube and sphere are special, they do not use caps. This makes them best for starting complex polygonal shapes such a heads).

Plane

Delete the cone and create a Plane object. You may have to move the camera back for a complete view.

The plane is useful, for example, as a picture inside a frame.

Others

The Landscape object is used to create an uneven terrain, such as mountains, based on mathematical algorithms (an algorithm is a defined as a rule, or set of rules, specifying how to solve a problem). The Relief Object lets you to control the shape of the terrain using a texture map in a material. The Human Figure is only for the purpose of practicing techniques such as learning how to use bones. It has no use as an object for scenes you may build.

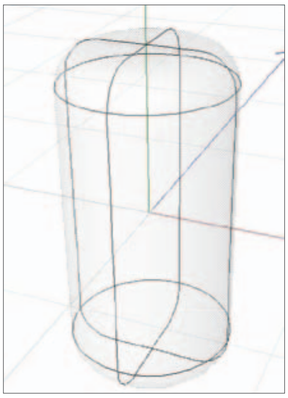
5. Displaying objects

You have already had some practice changing the display mode. Display modes affect how the objects appear in the editor, a few affect how the object is rendered in the editor but none affect the final render to the Picture Viewer.

Delete any existing objects and return to the default perspective view, View / Frame Default. Create a cylinder and move the camera in to enlarge the object in the view (shown at right). Rotate the view slightly so you can see more of the top. Now switch the display mode to Isoparms.

The lines displayed show a cross-section and profile only. No matter how many segments you have specified in the dialog, the cylinder will still appear this way.

Switch the display mode to Lines.



This still retains the Isoparm setting but no longer displays the surfaces. This is very fast to refresh and useful when lots of objects need to be displayed and moved at one time.

Add a Fillet to the cylinder and note the change.

Still, only the cross-section and profile. Notice that you can still adjust parameters interactively.

Try the Box display modes.

All you see is a rectangular box that encloses the object. This is very fast to refresh and useful when performing motion tests of very large or complex scenes. Even though you cannot make out the object's shape this mode will sometimes be sufficient to see how the objects move relative to one another while allowing for realtime preview speeds.

Return to Gouraud Shading (lines) and Wireframe display modes. Change the resolution of the object in the editor window to low, Display / Level of Detail / Low.

The cylinder still retains the parameters you set in the dialog (default values in this case) but these are only used for the final rendering. This is another way of reducing the Attribute Manager's amount of computation and allow faster redraw of the view.

If it is not already, uncheck Options / Level of Detail / Use Render LOD for Editor Rendering.

Render.

The object renders at the lower resolution.

Enable (check) Options / Level of Detail / Use Render LOD for Editor Rendering. Render.

This will force the editor render to use the same settings as those for the final render (to the Picture Viewer). This lets you model and position objects rapidly at lower resolution (*level of Detail = Low*) while still being able to do high-quality test renders in the editor window (*use Render LOD*).

There are also methods of displaying a specific object using one display mode while other objects in your scene use a different mode. For example, you could display the entire scene in wireframe while the object you are currently modeling is in Gouraud Shading. This is done by adding a Display Tag to your object which lets you apply Gouraud Shading, overriding the setting in the view window for that object only. You can try this out now if you have time but it will be a long time before our scenes are complex enough to require this feature.

6. The Object Manager

You should already have a pretty good feel for what the Object Manager does and how to use it. Here you will be introduced to a few more aspects of object organization and practice some techniques to speed up workflow. We will begin with your first hierarchy.

Delete the cylinder, return to the default perspective view and create three cubes. Name the cubes Cube 1, Cube 2 and Cube 3 by double-clicking on their names and typing text into the input box.

You can also change object names in the Basic Tab in the Attribute Manager.

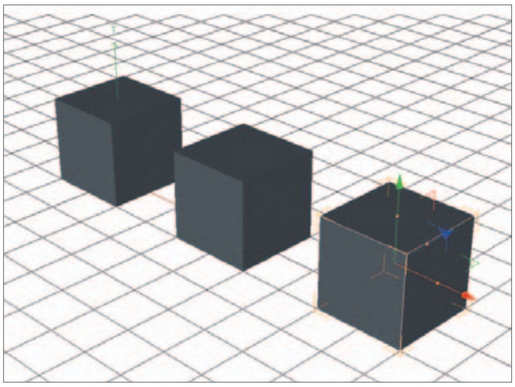
Note that the text in the input box is automatically selected when you open the box. Any typing will immediately replace this text. However, since you want to reuse part of the existing text, click at the end of the text before typing. The grayed selection area disappears and the cursor appears at the end of the existing text allowing you to add to it without replacing it. Switch to the front view using the function keys.

Learn the order of the views from *F1* through *F5* (perspective, top, side, front, four-view). Again, do **not** use the *Camera* menu in the view nor the icon in the view to change viewports.

Move the camera so that the cubes face toward the left side of the view. Select Cube 2 **and move it along the positive X-axis until it is a cube's width away from** Cube 1**. Check the Coordinates Manager (or the Coord. Tab of the Attribute Manager), the X value should read about 400.**

Remember, to move along one axis only, either lock the other two axes and move or grab and drag the appropriate axis.

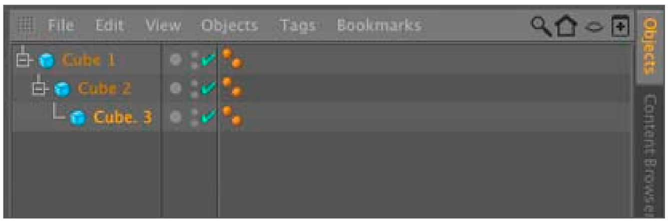
Move Cube 3**, also along the positive X until positioned at about 800); return to the perspective view and move the camera until you have framed all the cubes in the scene with some room to spare (see image at right).**



Next you will build a hierarchy consisting of the three cubes with Cube 1 at the top level.

Drag Cube 3 into Cube 2 (as you drag the object up over Cube 2's name release the mouse when the cursor shows the down arrow). Drag Cube 2 into Cube 1.

Next you will begin to see the uses of hierarchies and the Parent-child relationship.



Select Cube 3 and choose the Rotation Tool. Rotate Cube 3 freely.

No surprises here.

Undo any rotations and select Cube 2. Rotate freely.

The Child's motion is linked to that of the Parent as if they were a single object - a very important feature.

Undo and try the same with Cube 1.

The Null object

Null ("nothing") Objects can be used to group other objects, which helps organize and clean up the Object Manager. They also have other uses that are indispensable to animation. Here are the first of these.

Undo any previous rotations; Move Cube 1 **about 300 units in the positive X direction (using the Coordinates Manager).**

The two Children should move with their parent, Cube 1, maintaining the same distance between each of them.

Create a Null Object, Create / Object / Null. Drag Cube 1 into the Null Object. Select Cube 1 and the Rotation Tool; Rotate freely.

This should behave like before, rotating around Cube 1's axis at the center of Cube 1.

Undo. Select Null Object and rotate.

This is an important ability, rotation (actually revolution) about some point which is not the object's own center. We could have moved Cube 1's center (using the axis tool), without moving the cube itself, to the World origin (or any other place) but then we would have lost the ability to rotate it (and its Children) about its own center as well. You saw an example of this use of a Null Object in the introductory tutorial. The Null Object was located at the corner of a cylinder which was a Child of the Null Object. The Null Object rotated to make the cylinder topple over, then the cylinder rolled away by rotating about its own center (as well as being translated).

Other tags

You have had some practice applying tags to objects and adjusting them. We will look at one last tag here, others will be introduced later. Remember, tags are found under the Tag menu in the Object Manager.

Select Cube 2 and apply a Protection Tag to it. Try to move it in the view.

This is just what you would expect, but be careful, it will still move with the Parent.

While still in the editor window, click on Cube 1 to select it.

A word of caution: If you move the mouse at all while trying to select an object in the viewport you will not select the new object but will instead move the currently selected object. In general, it is best to select using the Object Manager.

Rotate Cube 1.

You will notice Cube 2 now moves along the rotation of Cube 1, even with the Protection Tag assigned.

Tags can generally be moved freely or copied (*control+drag*) between objects.

Drag the Protection Tag from Cube 2 into Cube 1, keeping it in the third column of the Object Manager. Try some rotations of both cubes and note the results.

This not only saves time but preserves any parameters associated with that tag which you may have previously set. You will sometimes find it convenient to drag a Tag onto a Null Object for temporary storage while you work out some problem and then drag it back when you are ready.

Multiple selections

CINEMA 4D lets you select and manipulate more than one object at a time. This is especially handy when animating but also has advantages during modeling.

Delete all current objects. Create a cube. Create another cube by Ctrl + clicking on the new cube. Drag down in the Object Manager until the cursor displays a left facing arrow and release the mouse. Do the same with the new cube to create a third cube.

Although this technique is not too important with default primitives, this is the method you should use if you need multiple copies of an object you spent a lot of time customizing. All objects duplicated in this way are identical, including all tags, textures, animation tracks and so on.

Drag the three cubes to different positions on the screen. Change the Size X value of one of the cubes to 100. In the Object Manager, select each object with a Shift+click.

Shift+clicking adds to the current selection so multiple objects can be selected. Make sure the Object Tab is visible in the Attribute Manager and study the values.

You should note that the *Size .X* input box is empty. This happens anytime multiple selected objects have different values for a given parameter. Conversely, when a number appears in the input box, all selected objects have the same value for that parameter.

Enter a value of 300 into the Size .X box.

All selected cubes should show the update (did you remember to perform some mouse or keyboard action to cause the value to be entered?)

Enter a value of 100 for Size .Y.

This method works whether the input box was empty or contained a common value. If different types of objects are selected simultaneously, only the parameters they share in common will be displayed in the Attribute Manager. For example, if you select a cylinder and a cube, only the "Basic" and "Coord." tabs will be displayed allowing you to, for example, move both objects to the same X position.

Grouping objects revisited

Finally, here is another way to group objects. Although it is not as versatile as the drag-and-drop method, you may find occasion to use it.

With the three cubes still selected, click on empty space in the right column of the Object Manager.

This makes the Object Manager the active window so it knows where to apply a keyboard command.

Hit the Alt+G key or Option+G key (on a Mac).

You should have produced a new Null Object with your three cubes inside. The center of the Null Object is at the geometric center of the three cubes. If you want the Null Object Object's origin to be at a specific position, it is better to create the Null Object independently, then drag the other objects into it. You will also need to use the drag & drop method if the objects are to be arranged in any other way than to be all at the same level within the Null Object.

Now you should have some understanding of primitives, parameters and how to work with objects in the Object Manager and in the Attribute Manager. Although this is a lot of information, it will become easier with practice.

Primitives Practice

These exercises will give you some practice in using primitives and adjusting their parameters.

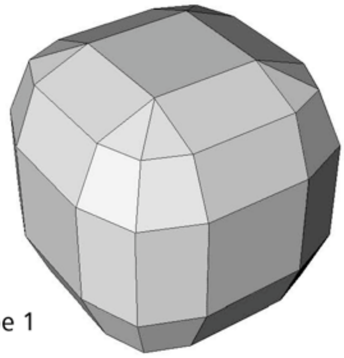
Open CINEMA 4D. Make sure you are in the perspective view (using a single viewport), with the camera focus to 80. Create a Null Object, name it "objects" and hide it. This is where you will store your objects as you finish each and move on to the next. Set the display mode to Gouraud Shading (lines) with Wireframe. Save the file in the "Intro Tutorials" folder as "Primitives Practice". Save often.

Hiding the Null Object will hide each of your objects as you drag them into the it. Each object must not be hidden individually.

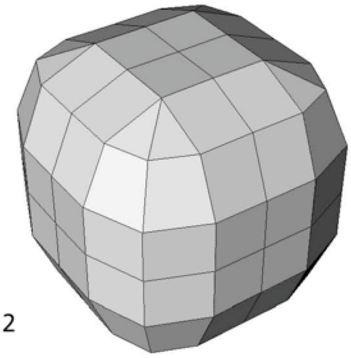
1. Parameter practice

Each of the following is shown with the *Phong* Tag deleted. Create each object and name it as indicated to the left or right of that pair of figures. Make sure to match the number of segments shown in the objects.

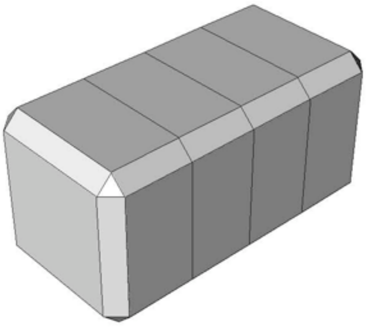
Cubes:



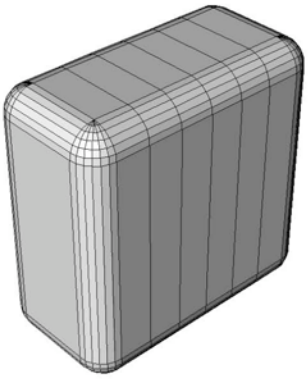
Cube 1



Cube 2



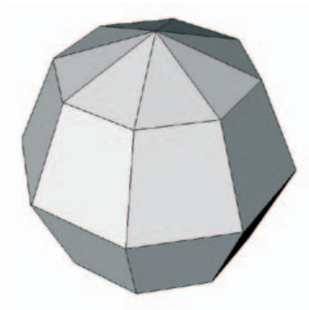
Cube 3



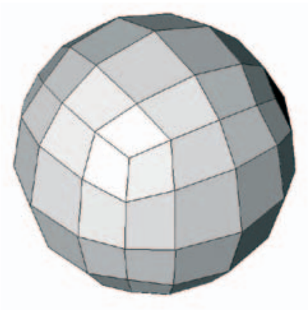
Cube 4

Spheres:

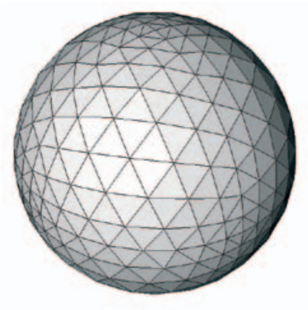
Sphere 1



Sphere 2

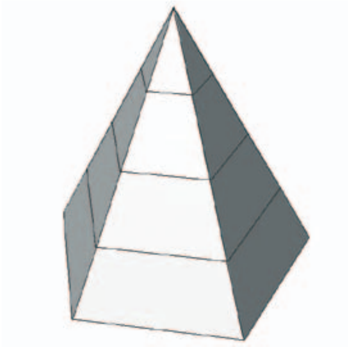


Sphere 3

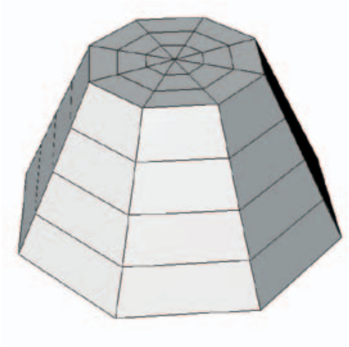


Cones:

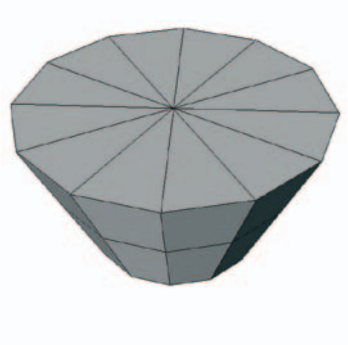
Cone 1



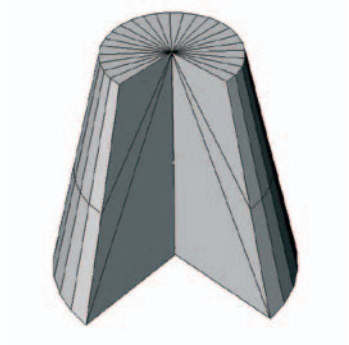
Cone 2



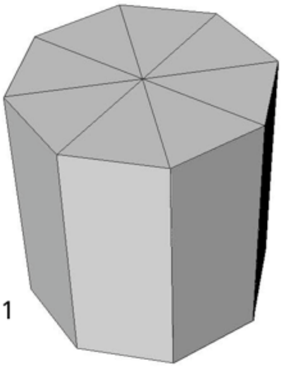
Cone 3



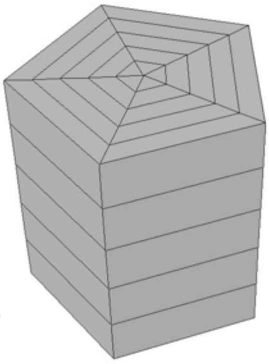
Cone 4



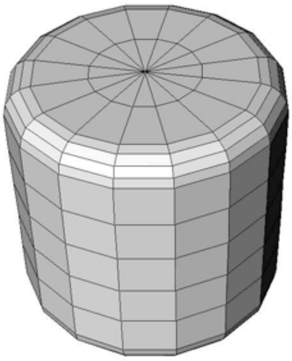
Cylinders:



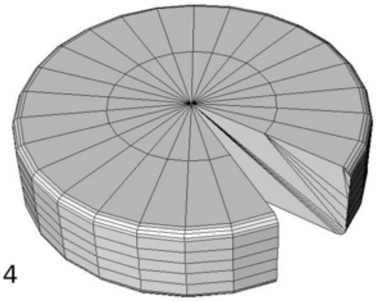
Cylinder 1



Cylinder 2



Cylinder 3



Cylinder 4

2. Matching

Each of the following has (at least) two primitives which can be used to create the shape.

Cube match:

Create a simple cube using the Cube, Cylinder, Cone, and Sphere primitives. Store all four in a Null Object named cubeMatch.

Some of the primitives will result in cubes that are rotated 45°. It will take a bit of experimentation to get the same proportions for the sides (hint: the square root of 200 is 141.4)

Cylinder match:

Create the cylinder using three different primitives (do not use the Tube Object). Store them in a Null Object called cylinderMatch.

This should be getting a bit trickier. You will have to experiment with a new primitive.

Pyramid match:

Create the pyramid using three different primitives. Store them in a Null Object called pyramidMatch.

One of your pyramids will have a three-sided base. Also, one of the pyramids will be limited to having all edges of equal length.

Capsule match:

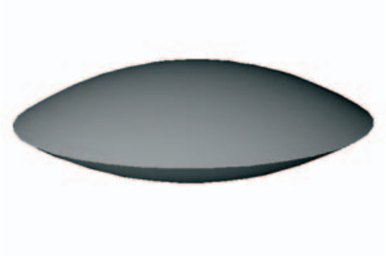
Create the capsule using three different primitives. Store them in a Null Object called capsuleMatch.

This one should not be too difficult to choose which primitives to use.

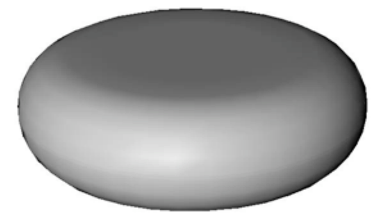
3. Find the Primitive

For the following shapes, choose the correct primitive and duplicate the object pictured. Name each as indicated and store all in a Null Object named findPrimitive. (Some of these require a little "cheating". If you set certain parameters to the minimum value to make the shape, such as the *Height* values in Pillow, artifacts may appear. If this happens, set the value a little higher than the exact amount necessary).

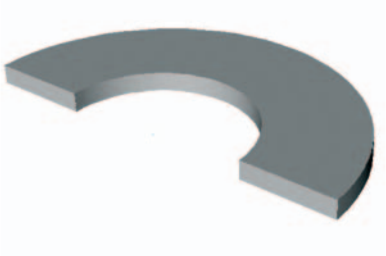
Lens



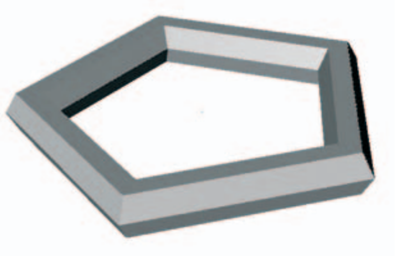
Pillow



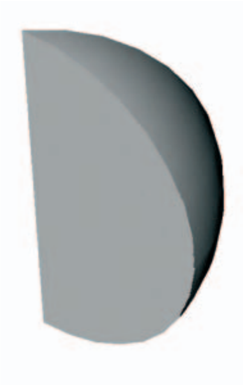
Arch



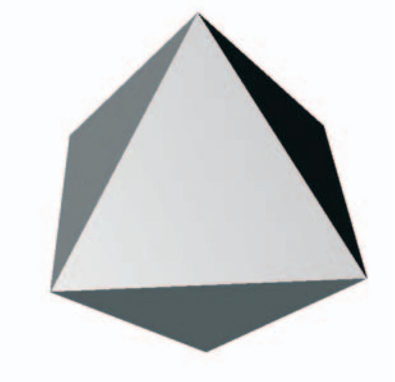
Pentagon



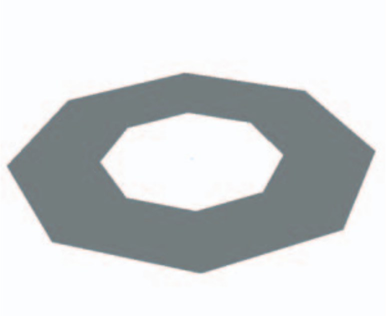
OSlice



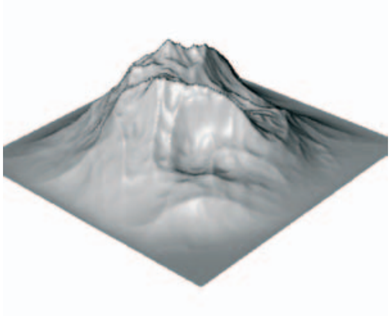
Crystal



Stop Sign



Mountain



You will often find use for primitives as parts of objects in scenes. Sometimes a single primitive will suffice, such as a sphere with the proper texture to create a basketball. Primitives may also be combined to make more complex objects. You could build a remote control for a television using cubes with filleted edges for buttons. Primitives could even be used to fashion a very cartoon-like character or a primitive cityscape, which appears convincing from a distance. However, more important is their use in fabricating a polygonal object for further modification at the polygonal level.

Splines and Generators

Overview

Occasional, primitives can be used as final objects. For most objects however, more complex methods of constructing objects are required. In a later lesson we will look at Polygon Modeling which offers the highest degree of control but also requires the most knowledge of tools and techniques.

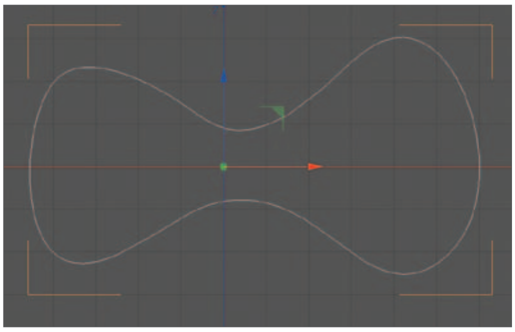
First, we will turn our attention to Splines, that are curved lines plotted through movable points and Generators. Generators can, in various ways, turn Splines into three-dimensional surfaces and can result in finished models of quite complex forms (right). One example of a Generator is the Lathe Object. Just like a lathe in a wood shop can be used to create a chess piece or a table leg, lathing can take a spline, which acts as the profile of the object, and rotate it around its own axis to make a 3D object.

Splines

Imagine several points in three-dimensional space. Now visualize CINEMA 4D calculating and drawing a curve using these points. This is a **spline**. There are several algorithms CINEMA 4D could use, resulting in different curves based on the same points. All of these Spline types are discussed in the CINEMA 4D Help files. We will focus on the most useful and important - others will be discussed later if necessary.

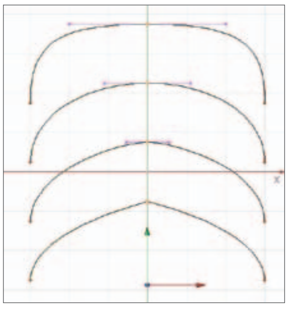
Linear Splines simply draw straight lines between the points. The other Spline types compute curves between these points. The most important, versatile and most used of these is the Bézier Spline. Bézier curves are common to most graphic programs. You may be familiar with them if you have used a "drawing program" such as Illustrator, Freehand, Canvas or Corel Draw.

Bézier splines



With a **Bézier Spline**, the curve passes through all the points. As a point is moved, the curve at that point is moved with it. Additionally, there are handles extending outward from each point called **tangents** (or handles), which allow you to control the shape of the curve as it passes through that point. As the tangent handles are rotated around the control point, the curve changes directions to remain tangent to the handles (that is why they are called tangents!) If the handles are lengthened, the curve becomes flatter at that point (note that the terms tangent and handle are used interchangeably).

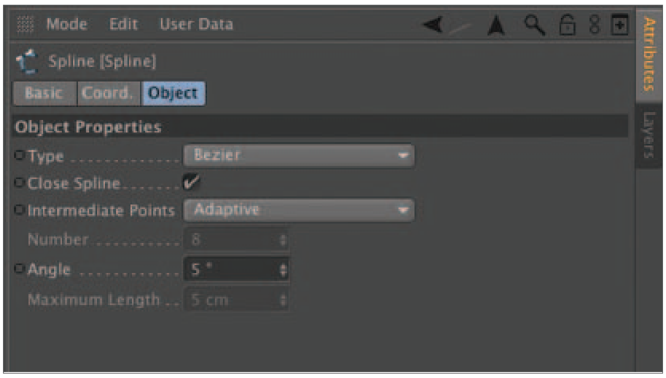
Tangents allow precise control over the shape of a curve using a minimum of points. The handles on both sides of a point may be moved simultaneously or independently of the other side. If both tangents are reduced to zero length, a sharp corner will result (unless the point is on a straight line between the two adjacent points). If all points on a Bézier had zero length handles, a linear Spline would result. Tangents can also be adjusted numerically via input boxes, accessed by double-clicking on any given point.



Bézier Splines are used to model object contours which can then be lathed, swept, extruded and lofted to produce a variety of complex models. CINEMA 4D also uses Bézier Splines for paths of objects when animated. As you move an object and record its position over time, CINEMA 4D computes a Bézier curve through the keyframe points. You can move these points and their tangents, just like any other Bézier, to modify the path the object follows over time. You can also draw a Bézier Spline and assign an object's motion to it. This would be highly useful in a Star Wars®-type scene when flying through an asteroid belt, or for having a car precisely stay on a road.

Splines can be open or closed. To close a spline, click on the first point after placing several points in order. This will also exit the Spline tool. Splines may also be closed in a check box in the Attribute Manager .

After creating a Spline you can edit its points using the *Point* mode along with the *Move Tool* or various other appropriate tools. Splines are important components of the 3D environment, so there are many tools for their use. A few important tools are introduced below, others will be discussed later as necessary.



Spline [Spline]

Object Properties

Type .... Bezier

Close Spline ...... ✓

Intermediate Points Adaptive

Number 8

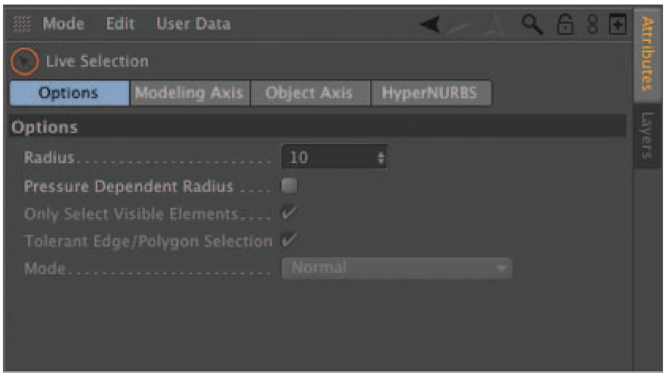
Angle .... 5 °

Maximum Length .. 5 cm

Useful tools and operations for points

- **Selecting points:** There are four selection tools available from the left side of the upper Command Palette (the arrow icon). These are used to select groups of points or polygons. The *Live Selection* Tool selects points as you click and drag over them. The others select regions of points when you drag a boundary around them. These tools have parameters accessible in the Attribute Manager.

The *Only Select Visible Elements* parameter affects whether the tool will select points on the backside of the object as well as those on the front surface.



Live Selection

Options

Radius 10

Pressure Dependent Radius .... □

Only Select Visible Elements.... ✓

Tolerant Edge/Polygon Selection ✓

Mode.......................... Normal

You may also add points to your current selection by holding down the *Shift* key while selecting additional points. Or, to subtract points from the current selection, press the *Control* key while selecting the points to be removed.

- **Moving points:** Once points have been selected, drag the mouse anywhere in the editor window (with the *Move Tool* enabled) and your selection will move as well. If you wish to rotate or lengthen a tangent click and drag directly on its handle. A very important time saver when working with points is using the *Space* bar to toggle between the *Selection Tool* and the most recently used other tool such as the *Move Tool*.

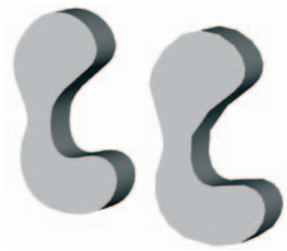
You can also select points directly with the Move Tool but you must Shift+click individually on each point you wish to add to a selection. This saves time if only a few points need be selected. A Shift+click on an already selected point will deselect that point.

- **Deleting points:** Select the points you wish to delete and press the *Delete* key. Who said this has to be difficult?

- **Adding points:** With the *Move Tool* selected, Ctrl+*click* on the Spline at the location at which you want a new point to appear.

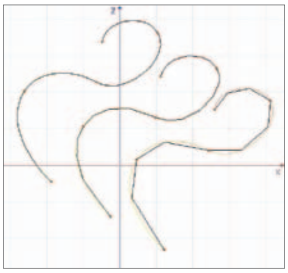
- **Tweaking points:** double-click on any point to open a dialog window which offers precise numerical control over its position and the tangent parameters.

Interpolation



The surface of a primitive is made up of small, flat surfaces, the number of which is determined by the number of segments. Likewise, Splines are composed of a series of short, connected lines as defined by the number and distribution of Intermediate Points. This is called **interpolation** and is adjustable in the Attribute Manager under the name *Intermediate Points*. The number of Intermediate Points on a Spline is directly related to how an object containing that Spline will be subdivided into polygons. In the figure at the right, both objects were made from the same Spline but the object at left has more interpolated points and therefore appears to have a smoother profile. Just like segmentation of primitives, more Intermediate Points results in a smoother surface but higher rendering times and larger memory requirements.

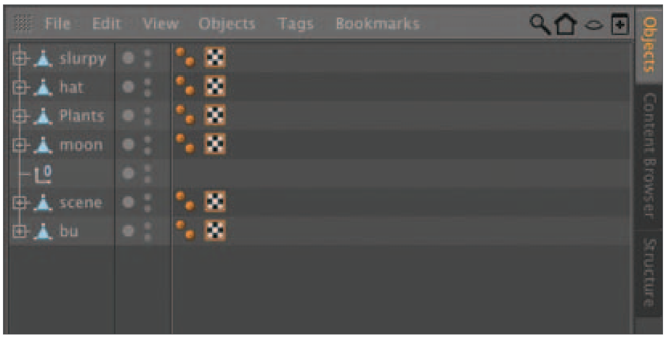
Although there are four types of interpolation available, almost all of your Splines should use *Adaptive* interpolation (this is the default method and generally the most efficient). As CINEMA 4D places points, it moves along the curve you have drawn until it has veered the number of degrees set in *Angle* then places the next point. Adaptive interpolation places points where they are most needed: More in tight curves, fewer in the straighter regions. For finer segmentation, use lower values of the *Angle* parameter. The three Splines to the right are identical, except for the Angle value. The values used, left to right, are 5°, 20° and 50°, respectively. As you can see, the default value of 5° produces a pretty smooth curve, good enough for most uses.



Here is the reason for interpolation. Although the Splines you draw look incredibly smooth on the screen, it would take forever to calculate a 3D object to this degree of smoothness. CINEMA 4D breaks the curve into short, straight-line segments between each of the control points you have created. Thus it can make polygons based on each segment when it is time to render. Adaptive interpolation is like driving down a curvy road placing stakes to get an idea of the curvature. You drive through a given point. Then, once you have turned a set number of degrees (say 5°) to the right or left of the direction of travel when you went through that point, you place a new stake to mark your next point. You keep doing this until you have traveled the length of the road. You then tie a rope from stake to stake along the road. When viewed from above, your rope line gives a pretty good approximation of the road's path by breaking it up into little segments. If you were to place a stake every time you turned only 2°, you would use many more stakes and would have a better fit to the actual curvature of the road. Of course this would be more expensive in terms of time and energy.

The other interpolation types are for specialized uses. None will give you straight lines between the points you have set in your spline. The object to the right was made from the same Spline as the objects pictured above but with interpolation set to none. *Uniform* is useful when animating a sweep, such as a name being written across the screen, so that there is a uniform amount of movement from frame to frame. *Uniform* is also useful when creating a Spline used for an animation path requiring uniform movement over time. The *Natural* interpolation method is used for rail Splines for sweep objects.

A note on backing up your work:



Splines are generally used in the creation of more complex three-dimensional objects. As such, they are often eliminated as the construction process continues and the object is converted to other forms. You may get far down this path and realize you need more or fewer polygons which depended on the number of Intermediate Points of the original spline. For this reason and others, it is a good idea to save copies of the intermediate states of your models. As you model objects, you should get in the habit of *Control+dragging* copies into a Null Object every so often.

A very important technique of file management is to backup your entire scene at different stages of development. For example, a project folder may contain the files "Robot_1.c4d", "Robot_2.c4d" and "Robot_3.c4d". Complicated, long-term projects may have 50 or more of these back-ups, although earlier versions can be purged when you are satisfied with the most recent versions.

Once again: **Back-up objects** within a file and **back-up files** by saving the newest version. Back-up often.

Spline primitives

There are several **Spline Primitives** available in the Command Palette. These behave just like user-created Splines, except they cannot be directly edited in point mode. Like other primitives, they have editable parameters accessible in the Attribute Manager.

Spline Primitives offer a quick way to accurately create such useful Splines as circles and 2D polygons. If you require further control they can be *Made Editable,* that will convert them into Splines whose points can be edited in the same manner as a Spline created from scratch.

One special Spline Primitive is the *Text Spline* which lets you enter text that can be used with Generators to create logos.

Generators

Generators are used to convert Splines into three-dimensional objects. Like primitives, Generators are higher order objects in that they have parameters which can be adjusted and animated. This is in contrast to polygon objects which must be manipulated with tools or modified by moving individual points. Generators can be used to create finished objects such as a lamp base, soda bottle, picture frame and even more geometrically complex objects such as a boat hull.

Objects made from Generators may also be converted to polygon objects to be further modified using other tools. Polygonal objects allow more variation and precision of control such as for the shaping of a mouth or making that mouth smile in an animation. However, higher-level objects offer ease of control over the entire shape. For example, movement of a couple of points in a Spline used for the cross-section of a picture frame will result in the entire frame being given a new profile. If this were a polygon picture frame, moving a few points would merely distort that region of the frame.

We will discuss the types of Generators that affect Splines to produce 3D surfaces in this section (later we will look at another type of Generator which does not affect Splines but rather polygon objects). All of these Generators result in NURBS (Non-Uniform Rational B-spline) objects named for the type of curve they calculate. In each case, a NURBS Object is created from the menu or Command Palette and a Spline is placed into the NURBS Object in the Object Manager (making the Spline a Child). You may still, at any time, modify the Spline using the Point mode. Following is a survey of these NURBS Objects (Generators). More detailed information can be found in the Help files.

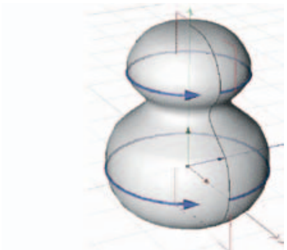
ExtrudeNURBS

EXTRUDED
TEXT

Extruding will take your Spline and move it along an axis of your choice to create a 3D object with straight sides. The amount of subdivision between the faces is adjustable in the Attribute Manager. The subdivision around the perimeter of the object is determined by the points and interpolation of the Spline.

Extrusion works well for anything with straight, parallel sides - like a cookie-cutter shape. Examples include a fluted (grooved) pole used in old floor lamps, a gear, an oval mirror, a guitar body, a baseboard and solid 3D text. Like all NURBS Objects, extrusions can have capped ends which may be rounded. This will be discussed at the end of this section.

LatheNURBS



Lathing takes a Spline and rotates it around the Y-axis to create a circular object when viewed from the top, and has the profile of the Spline when viewed from the front or side. Since rotation is always around the Y-axis, it is best to always draw your Spline in the front view. With Lathe Objects, the profile subdivisions are controlled by the Spline interpolation. The number of subdivisions around the lathe is set in the Attribute Manager.

You can also lathe less than 360° using the *Angle* parameter, for example when creating a cake with a slice cut from it. The rotation can be moved vertically using *Movement* to produce helices, and can be scaled as it is rotated using *Scaling* for spiral effects.

Along with chess pieces, lamp bases and even snail shells, you can create objects like donuts and springs. This is done by lathing a closed Spline around an axis that is outside a closed spline.

SweepNURBS

Sweeping requires two Splines, one of which is used as a **profile** (cross-section) that is swept along the second Spline - the **path**. The profile is the first Child Object under the Sweep NURBS Object, the path is the second. The Z-axis of the profile will follow the path, so you should generally create your cross-section in the XY plane (front view). The subdivision of a sweep object is based on both Splines.

The object can be tapered using *Scaling* and the cross-section can be rotated by *Rotation* as it is swept along the path. The *Growth* parameter is used when animating to allow your object to "grow" along its path over time like text being written across the screen or a "light saber" extending itself to full length.

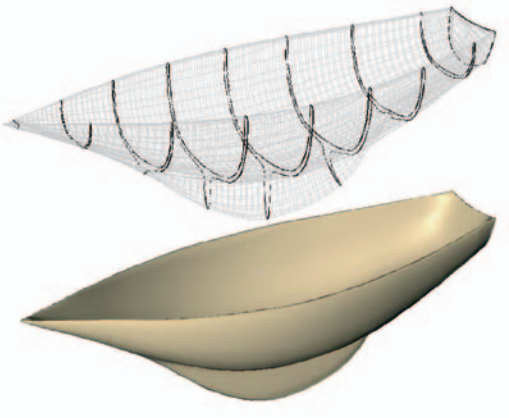
A third spline may be added which acts as a rail. This is used to modify the cross-section of the object along the path; many of the sweep's parameters in the Attributes Manager refer to this.

Sweeps have numerous uses beyond the obvious bent wire (i.e. paperclip) look. Using a rectangle as a path, an appropriate cross-section will result in a picture frame. Since splines can be animated, a sweep is also a good choice for a whip in action.

LoftNURBS object

Lofting takes many cross-sections and uses them as "ribs" over which to stretch the three-dimensional surface ("skin"). The major constraint is that all of the Splines used as cross-sections must have the same number of points. In practice, this means creating one Spline and copying and modifying it to produce the rest.

Notice in the Attribute Manager that both U (cross-section) and V (from Spline to spline) directions have their subdivisions controlled here. The interpolation set in the Splines does not matter for lofted objects. The *Loop* parameter will connect the last segment to the first for donut-like objects. *Subdivisions per Segment* will divide the V direction into the same number of subdivisions between each pair of Splines. When unchecked, the subdivisions will be evenly spaced over the entire object.



Lofting is very versatile and can be used to easily create objects such as boat hulls. Many organic forms are best created using lofting. Even a human face can be lofted, but it would not be very easy to animate. Other modeling methods are generally used for humans and animals. Slurpy, pictured at right, was created using only lofted Splines in an earlier version of CINEMA 4D.

Bézier NURBS

The Bézier NURBS creates a plane with points which, when moved, will cause the surface to pull up and out of the plane in a smooth curve. The points act as if connected to a plastic fabric (the surface) by rubber bands.

Although not used commonly, the Bézier NURBS is ideal for creating such objects as sailboat sails.

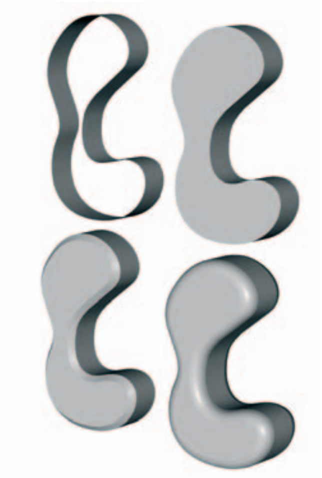
Caps and rounding

NURBS Objects like "extrude" and "sweep" move a Spline along a path and leaves open ends. These ends may be covered with flat *Caps* and/or rounded to various degrees. This is defined by the selections made in *Start* and *End* in the Caps Tab of the Attribute Manager.

Fillet defines the contour of the transition from the sides of the object to the caps. The *Radius* defines how rounded the edge will be and *Steps* defines how many segments, or subdivisions, the rounding will be made up of. *Steps* works just like the *Fillet* parameter of the *Cube* Primitive; a value of 1 will result in a chamfered (beveled) edge, a flat, 45° transition between side and cap. A value of 5 will result in a good rounded appearance, suitable for most purposes: larger values may be needed for larger, softer curved edges if seen close up.

You will find your model ballooning in size to accommodate the rounding unless you check *Constrain* (in some cases, with this parameter selected, you may notice rendering artifacts using large rounding values. This is especially a problem with text).

When the caps are calculated, the result is usually a number of triangular polygons on the end faces. These can sometimes cause problems in certain rendering situations such as the surface of a mirror. In cases like this or when you need to deform the face in point or Polygon Mode, the *Regular Subdivision* parameter can be applied. This divides the cap surface into a mesh of rectangular polygons.



Splines and Generators Tutorial

By now you should have an idea of the significance of points in modeling. In fact, points are the core element of an object's geometry with regard to how an object is calculated. Points define the corners of the polygons which make up the surfaces of all objects. This is most obvious when editing polygon objects, but is also true for parametric objects that are to be converted to polygons during rendering. Points are also the essential elements of Splines, controlling the location at which a Spline appears, as well as its curvature. First we will take a look at how Splines are constructed and edited, then see how they are used by Generators to make complex surfaces which can, in many cases, result in finished objects.

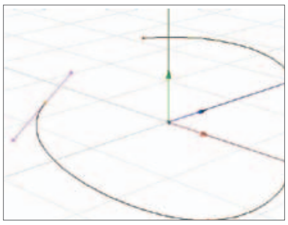
1. Spline intro

Although there are several types of Splines available in CINEMA 4D, by far the most useful, versatile and important is the Bézier (pronoun**ced “bez-ee-ay”**) Spline. With a Bézier, the Spline passes directly through the control points and these points have tangent handles which control the shape of the curve near that point.

Open CINEMA 4D and adjust the window; Change the focal length of the Editor Camera in the perspective view to 100. Now, move the camera back until you can see all of the World axis. Save the file as “Splines Tutorial” to your “Intro Tutorials” folder. Save often as you work through the tutorial.

How does CINEMA 4D know where in 3D space you want a point to be placed when creating a spline? Most often you will want all the points of a Spline to lie on the same plane. Even in rare cases where they do not you will usually create the Spline in a plane and then modify it.

While in the perspective view, create a Bézier Spline object by clicking on the Bézier Spline icon in the toolbar. You will not yet see anything in the Object Manager but the cursor will change its appearance in the editor window. Click and drag in the editor window.



You have now created your first point. The farther you drag the cursor, the longer the tangent will be at that point, resulting in a flatter curve.

Release the mouse button. Move the cursor to the location at which you want to create the second point and click and drag again.

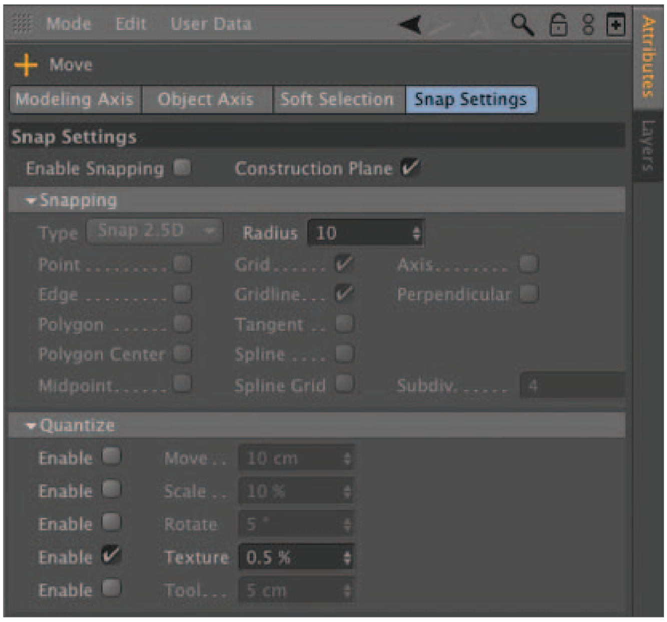
Now, as you drag the mouse around, you can see the effects of tangent length and direction on the shape of the curve.

Create a few more points in a somewhat circular direction toward the first point. Complete your Spline with a click on the first point.

This final click on the first point closes your spline, making it a closed loop. It also returns your mouse to edit mode. You will notice the *Point* mode is still selected and your points are visible, allowing them to be edited. Before move on, let's see where our spine actually is.

Rotate the view until you are looking at the Spline from the edge.

Notice your Spline was created on the ZX plane. To see why, click on the *Move Tool* and activate



Move

Modeling Axis | Object Axis | Soft Selection | Snap Settings

Snap Settings

Enable Snapping □ Construction Plane ☑

▾ Snapping

Type Snap 2.5D Radius 10

Point .... □ Grid .... ☑ Axis.......... □

Edge .... □ Gridline.... ☑ Perpendicular □

Polygon .... □ Tangent ... □

Polygon Center □ Spline ..... □

Midpoint....... □ Spline Grid □ Subdiv. 4

▾ Quantize

Enable □ Move .. 10 cm

Enable □ Scale .. 10 %

Enable □ Rotate 5 °

Enable ☑ Texture 0.5 %

Enable □ Tool... 5 cm

the Snap Settings Tab in the Attribute Manager. There is a check box next to the text *Construction Plane*. When this is enabled, your Spline will be created in the plane containing the grid lines. For any view except the perspective, such as a front or left view, the grid will lie on the plane of the screen.

If you were to disable the Construction Plane box your Spline will be created in the plane of the screen for a perspective view as well.

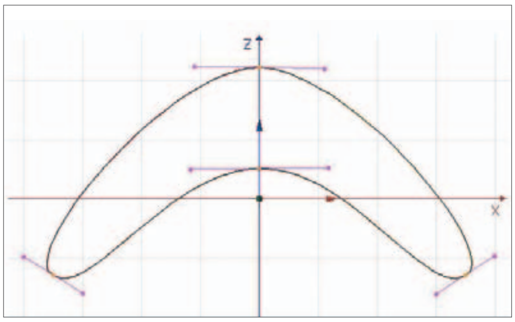
Now let's try a little Spline editing.

Since your first Spline was created in the ZX plane, switch to the top view so you are looking straight down onto it. With the Move Tool selected (the Point mode should still be selected as well), click on one of the Spline's points without dragging.

You have to be careful when selecting points on a spline. If you drag while clicking on them you will instead move the previous selection of points.

Click and drag somewhere in the editor window (but not over any points). Release when the point is positioned where you want it. Now click and drag on one of the tangent handles.

Experiment a bit to see how the curve varies as you manipulate its tangents.



Delete the Spline. While still in the top view, create a new closed Spline with four points. Try to create a boomerang shape using the Spline. Edit the points to refine your boomerang.

If you were to rotate the view you would see that all points and tangents are still in the ZX plane. Let's try adding a point.

With the Point mode still selected, Ctrl+click on the Spline where you want to insert a new point.

The new point appears and the curve maintains its previous shape. You can move this point and its tangents like any other point.

Delete the point by pressing the Delete key with the point selected.

Note that the curve has changed in shape. Tangents of neighboring points will be adjusted when you add a new point to maintain the shape of the Spline. However, when a point is deleted, tangents of neighboring points are left as they are, causing a change in the Spline's shape.

You may also create open Splines.

Delete your Spline and create another with three or four points. Do not click on the first point as done previously; click on the Move Tool.

Clicking on the *Move Tool* (or any other tool) lets CINEMA 4D know you are finished creating your Spline and lets you edit the points, if necessary. Now you have an open spline, editable just as any closed Spline can be edited. What if you later wish to close this spline?

Click on the Spline's name in the Object Manager.

This will replace the *Move Tool*'s attributes in the Attribute Manager with those of the Spline.

In the Attribute Manager, check the Spline's Close Spline box.

You may also open any existing closed Spline by unchecking this box. The Spline will open between the first and last points. These are the colored gradient in the Spline whose color runs from white at the beginning to blue at the end of the Spline. The order of the points can be changed but this is not usually necessary, except for advanced modeling situations.

Select one of the points on the curve. Hold down the Shift key while you click and drag on a tangent handle.

The handles can be adjusted independently of one another.

Drag the handle again without holding the *Shift* key.

The two tangents once again move simultaneously. You may align them precisely in the same direction again by using Mesh / Spline / *Equal Tangent Direction*. You may also make their lengths equal by using Mesh / Spline / *Equal Tangent Length*. These can both be accessed via a pop-up menu.

With the point still selected, right-click in the editor window. Move the cursor to *Equal* Tangent Direction and click.

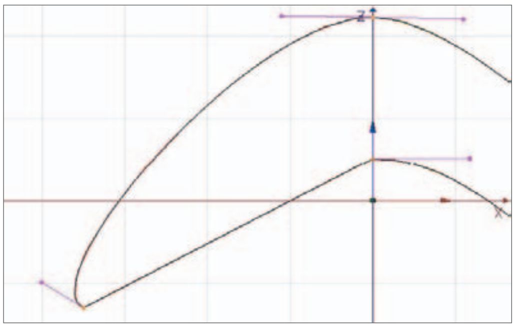
Sometimes you may want tangents to have zero length so that you can create a sharp point (cusp) on the Spline.

Using the pop-up menu again, choose Hard Interpolation.

There may be occasions where you want just one of the tangents controlling a point's curvature to be zero length. This has been done to two points in the figure at the right to create a straight line segment between them. To do this you can double-click on any point to open a dialog box which lets you precisely set values for the tangent handles as well as the point's position.

There are many other tools for editing points but we will discuss them as needed later.

You should see input fields in the Attribute Manager for adjusting how the Spline is calculated between the points you have created (*intermediate Points*, etc). If you were to change these parameters now, you would not notice any difference; a Spline is always drawn "perfectly" on the screen. However, these parameters do affect how the Spline is subdivided when used in modeling a surface. We will take a closer look at these parameters in the section on Generators.



2. Selecting multiple points

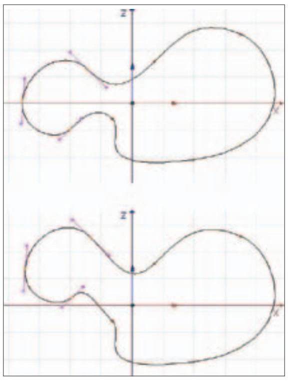
So far you have been working with a single point at a time. You will, however, often have to manipulate several points simultaneously. Here is how to select just a few points simultaneously.

Delete any existing Splines. Create a new Spline in the top view with 6 to 10 points. Select a point while the Move Tool is selected. While holding down the Shift key, select another point.

You will now have two points selected. You may add additional points to the selected points in the same manner. Remove points from your selection set in the same way.

With the Shift key selected, click on an already selected point.

That point should be removed from your selection.



Add two more points to the selection. With multiple points selected, click and drag in the editor window (remember not to click over any points or you will change the selection).

The selected points should now move around as a group. They may also be rotated and scaled as a group. With the group still selected, choose the *Scale Tool* and click and drag. Try the *Rotation Tool* as well.

This method of adding points to a selection is good for a few points because it can be done with the *Move Tool* without having to switch tools. More often, especially when selecting many points, it is quicker and easier to use the selection tools. These tools are also indispensable when working with polygonal objects.

Deselect all points. The best way to do this is Cmd+Shift+A (cmd+A, is the "select all" function).

Cmd+A and *Cmd+Shift+A* work on the current active window. They will work on objects in the Object Manager and materials if the Material Manager is currently active.

From the toolbar, choose the Live Selection Tool. Click and drag over several points you wish to select.

Notice that the cursor turns into a circle when this tool is selected. This is the radius of selection. Any point this circle moves over will become part of the selection. For some situations, you will want to adjust the size of this radius.

In the Attribute Manager, change the Radius to a value of 20. Select three or four points in the editor window.

Notice that the size of the selection radius has doubled. Also note that the previously selected points have been omitted from the selection set.

Select additional points with the Shift key pressed.

This adds to the current selection set. You can also remove points.

Click and drag over some of the currently selected points with the Control key pressed.

There are other tools for selecting larger groups of points.

Deselect all points. Choose the Rectangle Selection tool. Drag the marquee around a group of points.

Actually, it was not necessary to deselect the previous group of points above. It was done so you could better see how the tool works. If no modifier keys (*shift* or *Control*) are used, any new actions with a selection tool will replace the currently selected set.

The *Shift* and *Control* keys also work in the same manner with the other selection tools.

Deselect all points and activate the Free Selection Tool. Select some points by dragging a closed curve around them. Practice adding and removing from the selection using the Shift and Control keys.

There are other features of the selection tools as well as several other tools to help with selection of points and polygons. We will look at them later when we cover Polygon Modeling.

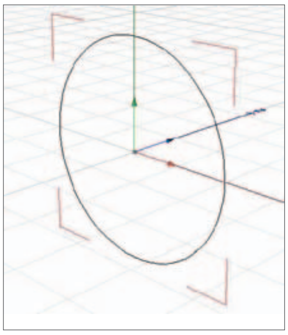
One last point before we complete our introduction to Splines:

Render your Spline in the editor window (Render Active View).

Nothing appears? This is because Splines have no thickness and therefore are not visible when rendered. They must be translated through 3D space to construct a visible surface. This is the function of Generators.

3. Parametric splines

There is a set of predefined Splines with adjustable parameters, similar to those of the primitive objects.



Delete your Spline and make sure you are in perspective view (with the Focus set to 80). Create a circle Spline from the main menu, Create / Spline / Circle or from the toolbar. Select the Point mode.

First, note the orientation of the circle in the XY plane facing the front of the World view. If you were to create a Spline Primitive in one of the dedicated views such as top or right, the Spline would automatically be placed in the plane of the view. Next, notice there are no points to edit, you must adjust this circle at the parameter level.

Make the circle into an ellipse which is half as high as wide.

You can do this using the *Ellipse* check box and the *Radius Y* parameter in the Attribute Manager. Notice, also in the Attribute Manager, how you can also change the plane of the circle’s orientation. This is the only way you can change the axis of rotation relative to a parametric spline. This is important when Splines are used with some types of Generators, as you will see later.

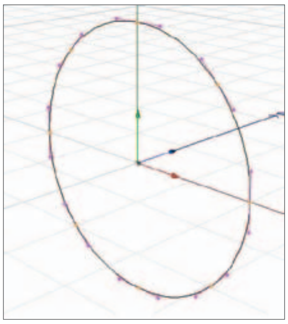
Suppose you want to create an egg shape. This could be done pretty quickly if we could start with a circle. Spline Primitives can be used as a starting point, then converted to normal Splines with points (“made editable”).

Return your ellipse to circular form. With your circle still selected, choose Mesh / Conversion / Make Editable from the main menu or by clicking on the appropriate icon ("two spheres"). Select the Point mode and the Move Tool. Select the top point and drag its X-axis upward to elongate your circle.

You could also lock the X and Z-axes and drag, but using the axis-drag method means you do not have to unlock the axes again for other manipulations. It also provides for numerical feedback as you drag. You must exercise caution when moving points on a spline. Make sure that they are not moved outside of the plane of the Spline. It is generally best to edit Splines in top, front or side views and not to use a perspective view.

Other Spline Primitives will be discussed as needed, or you can experiment with them yourself. Before we go on to Generators, here is a glimpse at a feature usually used on surfaces, but also works with Splines.

Restore the Spline to a circle; be sure all points are deselected. From the pop-up menu, choose Subdivide. In the Subdivisions input box, enter a value of 2.



More points but still a perfect circle. This will work with any Spline or section of a spline.

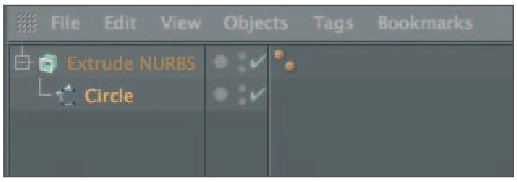
Using a selection tool of your choice, select any three consecutive points. Subdivide again.

Of course this is not necessary if you intend to leave the geometry as is. It only makes sense if you want to further adjust a shape. There are several other primitive Splines, all can be made editable for finer manipulation.

3. ExtrudeNURBS

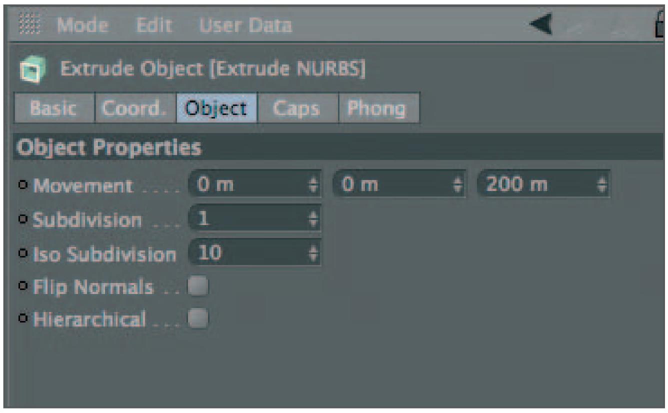
OK, here is where you begin to actually make useful objects.

Delete any current objects. While in the perspective view, create a Circle primitive. Make it editable. Create an Extrude NURBS Object, either from the main menu, Create / NURBS / Extrude Nurbs, or from the toolbar. Drag the circle into the NURBS Object to make it a sub-object.



You should now have a coin-like object. Let's make it thicker.

In the Attribute Manager, change the Z Movement to 200 (the three input boxes listed to the right of Movement are for the X, Y, and Z-axes).



Extrude Object [Extrude NURBS]

Object Properties

- Movement . . . . 0 m | 0 m | 200 m
- Subdivision . . . 1
- Iso Subdivision 10
- Flip Normals . . □
- Hierarchical . . . □

Your coin has been extended along the Z-axis. Although this object looks the same as the cylinder primitive we saw in the last tutorial it is very different. If you were to begin moving individual points of the circle around you would see the shape of the object change accordingly. The Extrude NURBS Object works like a cookie cutter, producing an infinite variety of profiles. However, the two faces (top and bottom of the cookie) are always parallel.

Choose the Point mode and make sure the object Circle is selected. Select the top point of the circle and move it downward along the Y-axis only.

Before we can do productive modeling with NURBS deformers, we must first suffer through some details of how NURBS Objects are modeled. (the following directions assume you are currently setup to use the HPB system. This can be set in the Coord Tab of the Attributes Manager).

Undo the deformation you just made to the Extrude Object. Select the Model mode and the Rotation Tool. Lock the P and B axes. Now, select the object Circle and rotate its heading in the editor and observe the changes to the object.

You will notice that the extrusion direction remains along the Z-axis of the Extrude NURBS Object (you can change which axis is used in the *Movement* input boxes of the NURBS Object).

Also observe that what is extruded along the Z-axis is not the circle itself but the projection of the circle onto the ZY plane of the NURBS Object. Simply put, the profile that is extruded is that of the circle as it appears when looking down the Z-axis of the NURBS Object.

Try rotating the circle's heading while looking at a top view (set to Gouraud Shading).

Notice the sides of the object always stay parallel to the object's Z-axis (which in this case happens to be the same as the World axis).

Now switch to the front view (also set to Gouraud Shading) and rotate the circle.

The NURBS Object does not even appear to have sides because you are looking directly down its edges with no perspective.

Now return to the perspective view and try rotating the Spline around its pitch and bank axes. When you have a feel for the effects of these rotations, return your circle to its initial orientation by entering values of 0 in the *H, P, B* input boxes in the Coordinates Manager or Attribute Manager.

If you entered values for Y *Movement* instead of Z in the Attribute Manager, the circle would have been extruded along the Y-axis instead. Looking along the Y-axis of the NURBS would make the circle (in its present orientation) look like a straight line. When you extrude a straight line the result is a flat plane.

4. Subdivision, NURBS and splines

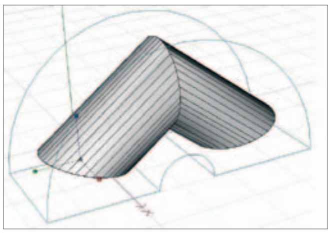
In the Object Tab of the Attribute Manager, with the extrude NURBS active, you will notice two other parameters with input boxes. The one named *Iso Subdivision* only defines how many lines are used to display the object in the editor when the Isoparms method of display is chosen in the viewport. It has nothing to do with the appearance of the object when rendered. The *Subdivision* parameter, however, will determine the number of segments between the front and back surfaces of the object along the extruded axis. In this case, along the Z-axis.

Even though the extrude NURBS is not a polygonal object, it must be converted to polygons during rendering. The number of polygons is determined by the *Subdivision* settings. If we display our object in wireframe mode we will see how the object will be subdivided. To unclutter our view of the basic geometry we will temporarily eliminate the caps so our object will appear as a hollow tube.

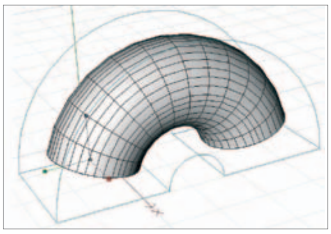
Be sure the Circle's H P B orientation to 0, 0, 0. In the Attribute Manager, change the Z movement to 1000 and, in the Caps Tab, set both Start and End to None. Change the display mode to Gouraud Shading (lines) with Wireframe. Move the camera in the perspective view so you can see the entire object. In the Attribute Manager, change Subdivision to 4.

These extra subdivisions are not needed if the object's sides are to remain straight. However, if you want to later use a Deformer (e.g. "bend") on this object, subdivision along V will make quite a difference. More subdivisions will result in a smoother bend. We will revisit this in the Deformers section. Subdividing might also be necessary if you intend to convert the object to a polygonal one (*Make Editable*) and may also affect texturing.

Bend with 2 subdivisions



Bend with 12 subdivisions



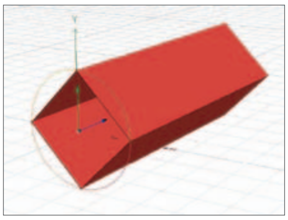
Reset the Subdivision to 1 and render in the editor.

Looks pretty smooth. Again, changing the subdivisions will not affect the appearance if our object is not deformed. However, what about subdivisions running around our object as opposed to along the extruded axis? This is controlled by the Spline used for the profile of the NURBS.

Delete the Phong Tag associated with Extrude NURBS. Render and observe the degree of smoothness in the curvature of the cylinder.

A close look reveals the subdivisions going around our extruded cylinder.

Select Circle and set the value of Angle to 20. Render.



Hope you remembered to go the Attribute Manager to change the value. Now the faceting is far more obvious as a result of a lower subdivision count. When you increased the Angle value, why did the subdivision count decrease? When using adaptive interpolation (interpolation refers to setting *Intermediate Points*), CINEMA 4D begins at the first point and travels along the ideal curve (it cannot just use the ideal curve that appears on your screen to actually make a 3D surface. This would result in far too many subdivisions and overwhelm the computational ability of your computer).

When CINEMA 4D has turned an angle of 20° traveling away from the first point, it places an intermediate point. It continues to do this every time it turns 20° from the previous point. So, smaller *Angle* values break the Spline up into more subdivisions, making a more precise fit to the Spline. As always, more subdivisions means more memory and longer redraw times. Again, with the *Phong* Tag the object would have looked pretty smooth on the surface. However, just as you have seen earlier with subdivision and primitives, the silhouette would have shown the facets.

Try an Angle value of 90. Render.

Now you have 90° between the sides of a square. In traveling around a circle, you will have turned 90° four times, thus four points.

The *Angle* value is not always adhered to strictly. If you were to subdivide the circle so it contained eight points instead of four and then set *Angle* to 90°, it would break the profile up into eight sides instead of four. *Angle* only applies to interpolated points. The calculated curve will always pass through the actual control points you have created in the editor (with Bézier Splines).

5. Caps and rounding

When a Spline is used to create NURBS Objects, the geometry is calculated by moving the Spline in some manner. In most cases, this leaves open ends that are then capped separately. The beginning cap is called *Start* and the other is called, surprisingly enough, *End*. You have control over some capping parameters, i.e. you have already turned the caps off for the above object. Let's see what else can be done to close the ends of NURBS Objects.

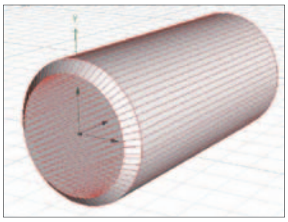
Return to an angle value of 5° for the circle's interpolation and set the Subdivisions of the extrude NURBS back to 1. Set Start and End to Cap.

The wireframe shows how the ends have been capped by polygons (you may notice this looks different from what you saw at the ends of the cylinder primitive).

Let's take a look at rounding those edges.

In the Caps Tab of the Attribute Manager, change both Start and End to Fillet Cap. In the Details Tab of the dialogue, change the Radius for both the Start and End to 40.

First, notice your object has gotten larger. You will also see the bevel around the edges where the caps meet the sides. Let's make sure adding the bevel keeps the object the same overall size.



In the Caps Tab, check the Constrain box.

If you want a curved edge, add more steps to the transition from cap to edge.

In the Details Tab, change Steps for both Start and End to 5.

This should be looking familiar, like the fillet parameter of the Cube Primitive.

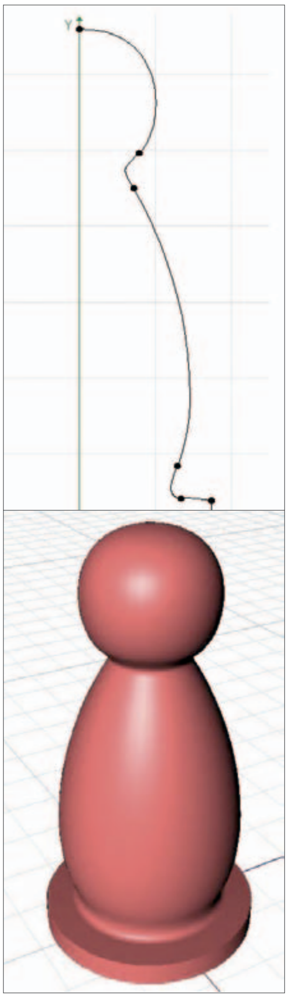
Same result: More steps => more smoothness => more memory => slower speed.

Rotate the view for study.

Five steps does a pretty good job for most purposes. Remember, adding a *Phong* Tag will smooth the surface quite a bit (but not the silhouette). Try some of the other Caps and Rounding options to see what they do.

6. LatheNURBS

An Extrude Object will push a Spline along the Z-axis. A lathed object rotates a Spline around the Y-axis of the Lathe NURBS Object. Until you get very comfortable with the program, always draw the Spline for a Lathe NURBS in the XY plane (front view).



Fold and hide your Extrude NURBS Object both to the editor and renderer. Switch to front view. Choose the Bézier Spline object.

Use the image to the right as a guide in creating your spline. The Spline pictured has eight points, the fewer the better for most purposes.

Create your Spline by placing points in the editor, beginning at the bottom, with the end point at the World origin. As you place each point (except the first three), drag to create tangents which will allow you to adjust the shape of the curve. Make sure you only edit this Spline from the front view so that the Z-axis values for all points and tangents remain zero. If you do not do this you will end up with distortion in your object. Since you want an open spline, click on the Move Tool after placing the last point.

Do not try to match the picture exactly, but do try to attain smooth curves (if we wanted to precisely match the shape pictured, we could import an image and use it as a pattern to trace over). The critical part, to avoid holes or little nubs at the ends, is to place the two end points right on the Y-axis (X = 0).

Select the top end point and set the X value to 0 (in the Coordinates Manager). Make the tangent as horizontal as possible so your lathed object will be smooth on top. Also set X=0 for the end point at the bottom.

Now the Spline will be lathed exactly around these two end points.

Create a Lathe NURBS and drop your Spline into it. Switch to perspective view and Gouraud Shading so you can rotate and view your object.

Something similar to a chess pawn has been created.

Rotate the view so you are looking down at the object at a steep angle. Select the Spline, make sure the Point mode is still selected. Select the top end point and move it away from the Y-axis along the positive X-axis.

You have probably opened a hole in your object. This is why you took pains to make sure the two end points were on the Y-axis (X = 0). You can move any of the points in the Spline while the NURBS Object is active and get immediate feedback as to the change in shape. However, be careful: Any moves not done in the front view (XY plane), or with the Z-axis locked, will probably result in the points moving out of the XY plane, which is not what you want for a lathed object.

Be sure to undo any moves you have made to the top point.

Now for a look at some other parameters of lathed NURBS.

Make sure the object fills the editor window and return to Gouraud Shading (lines). Note the amount of subdivision. From the NURBS dialog, change the Subdivision to 8.

Notice, in this case, the subdivision is around the object. This is like the extrude because it still is in the direction the Spline is being moved. The U direction is still along the Spline.

You should be able to figure out how to change the subdivisions along the vertical direction.

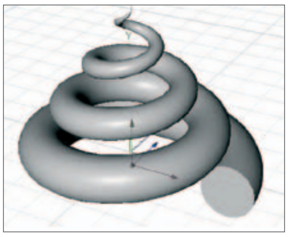
Reduce the subdivisions along the vertical direction (Y-axis in this case) by about one half.

Were you successful? (hint: Change Angle for the Spline's interpolation)

The *Angle* parameter of the lathe dictates how far around the circle our object will turn (taking a slice from a cake). *Movement* causes the Spline to be moved along the Y-axis as it is being rotated (as in coiled spring). *Scaling* changes the Splines size during rotation (making spirals). These have special uses as shown in the following example. Note that you might not have thought of using lathing in this manner; a major factor in successful modeling is learning how to "think outside the box", or how to use tools in imaginative ways.

Reset any changes you may have made to the Spline. Change the name of your Lathe NURBS Object to Pawn and reset the subdivision back to 24. Fold Pawn and hide it. While in the front view, create a circle Spline and adjust its Radius to 80. Move the circle to an X position of 300 using the Coordinates Manager (if the Point mode is still active, the Coordinates Manager can only move selected points. Make sure the Model mode is active to move the Spline object). Create a new Lathe NURBS Object, name it Snail and drop the circle into it. Return to the perspective view and rotate to view the object.

This basic shape is called a torus. There is also a torus primitive but it is much more limited. You could, of course, make the circle editable and move points so that the cross-section was not a circle. In this manner, you can make rings, simple tires, gaskets, etc. However, let's see what several lathe parameters together can do.

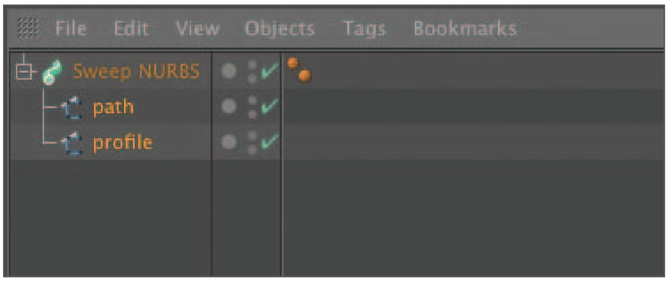


For Snail, change Angle to 1440, set Movement to 500 and enter 0 in the Scaling field.

The 1440 *Angle* value turns our Spline around four times (4 X 360°) while it is lifted by *Movement* and tapered (to 0%) by *Scaling* . The result is very rough. It is not very smooth as it goes around, the 24 subdivisions are now spread thinly around four rotations.

Make it smooth! Render it.

7. SweepNURBS

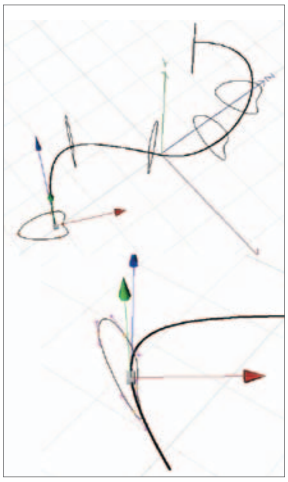


Sweep NURBS
path
profile

This NURBS Object requires two Splines. A contour Spline (or profile), defines a cross-section which is moved along a path spline, to create a rope or snake-like object, similar to an extrude along a curved path.

Fold and hide Snail. Create a new closed Spline with 4 or 5 points in the XY plane to be used as the contour (profile). Center the Spline around the World origin and make sure it is created on the XY plane. Name this Spline Profile. Adjust the size of the Spline so it is about 100 units wide by 100 high. Create a second open Spline in the ZX plane to be used as the path. Try an "S" shape about 500 units long. Name this Spline Path. Now create a Sweep NURBS Object and drop both Splines into it.

Be sure both Splines are at the same level within the NURBS Object and that



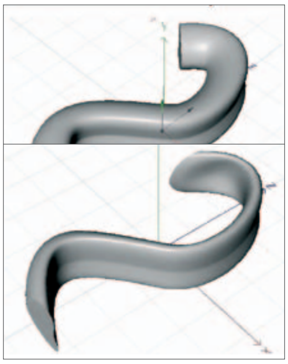
Profile is above Path.

Switch to the perspective view and rotate the view to see your object. Try moving points on the path.

Is it a bit difficult to select points?

Select the Sweep NURBS Object again and, in the Basic Tab of the Attribute Manager, check X-Ray. Switch to Gouraud Shading. Select Path again and move points and tangents.

This is probably the most common situation where you may not want a spline's points to all be in the same plane (imagine constructing a knot). It does not matter which direction the contour Spline (Profile) is facing. What matters is how it appears when looking down its own Z-axis (this is different from the extrude!) The shape as it appears looking straight down its Z is what will be swept along the path. As the contour is swept along the path, its Z-axis will always be kept tangent to the path (upper figure on right) unless the *Parallel Movement* parameter is enabled in the Attribute Manager.



The situation becomes different if you rotate the contour's axis (lower figure on right). When this is done, the Spline appears in the same position but has a different orientation to its own Z-axis. When you look down the Z-axis, you will see a different shape than before. This new view down the Z-axis will be the profile sweep along the path.

Select the Profile object and Enable Axis modification in the left hand tool bar and rotate Profile's heading until the sweep takes on a thin ribbon shape. Now deactivate the Sweep NURBS (by unchecking in the center column) so you can see the profile Spline and make sure Profile is still selected. Rotate the view so you re looking straight down the profile's Z-axis. This is what CINEMA 4D is sweeping down the path.

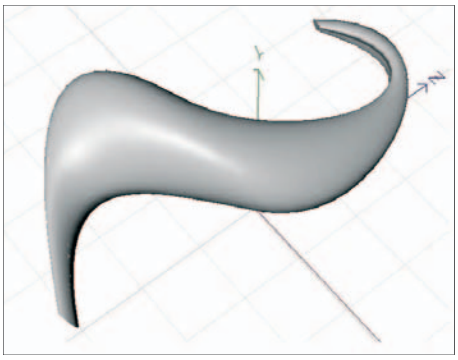
Your object's path remains the same but the cross-section along the path should appear like the view of Profile down its Z-axis, much thinner than before. To avoid this distortion, always create your profile in the front view.

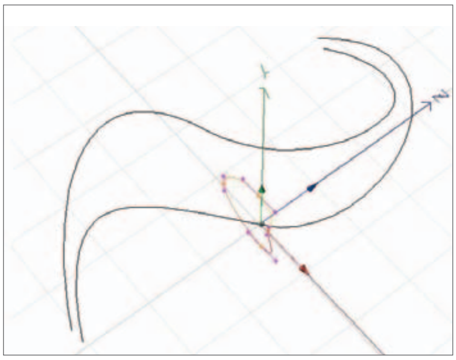
The *Banking* parameter (on by default) rotates the contour while it is swept so its Z-axis of the profile is always tangent to the path. If you want the Z-axis of the profile to keep a fixed orientation in space independent of its journey along the path, choose *Parallel Movement* instead.

Sweep NURBS may also use a third Spline called a rail which can cause the profile to vary as it moves down the path. Use *Natural* interpolation (*Intermediate Points*) for both the path and rail if you choose to do rail Splines.

Undo any rotations to Profile's axis. Change the Intermediate Points of Path to Natural.

Duplicate Path by Ctrl+click-dragging the Spline Path downward in the Object Manager until a left arrow appears. Rename this Spline Rail (your SweepNURBS Object should now contain three Splines, all at the same level). In the top view, move Rail over a little to the right and adjust its points so the spacing between the two Splines is uneven. Perhaps try to create a bulge in the center, with tapered ends. Reactivate the NURBS and view the results in perspective.



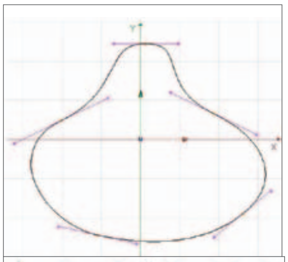


This is a powerful deformer. Play around with the Splines a bit while the NURBS Object is active. You may need to enable X-Ray to gain access to hidden points. Sweep NURBS are made up of Splines that flow in either direction so the subdivisions are controlled by the Splines, not by the NURBS Object.

8. Loft NURBS object

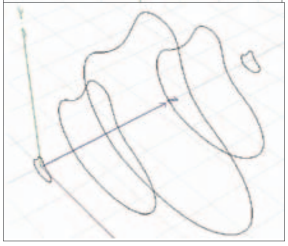
Lofting is very powerful. You can use any number of Splines to keep the profile changing as the surface is generated.

Fold and hide Sweep NURBS. Create a new closed Spline in the XY plane with about 6 points (at the right is a suggestion). Make the overall size about 400 by 400.



We want to duplicate our Spline four times using a method that will automatically number our copies sequentially.

Rename the Spline s. Control drag the renamed spline in the Object Manager 4 times, automatically renaming the new Splines s.1, s.2, s.3, s.4 and s.5. Delete the original spline. Move s.1 to a Z position of 600, move s.2 to 450, s.3 to 300 and s.4 to 150, all in reference to the Z-axis (watch the value in the Coordinates Manager for a guide as you drag or enter values directly). Switch to the perspective view and select both s.1 and s.5; choose the Scale Tool but lock the Z-axis. Scale to overall sizes of about 50 X 50.

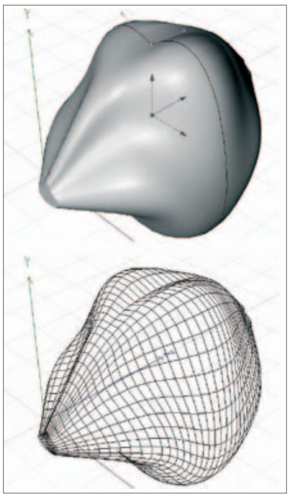


Had you not locked the Z-axis, scaling would have scaled the Splines in towards each other. This would not be a problem when scaling them individually (since they have no depth along the Z-axis) but is something to take into account when scaling multiple selections.

Scale s.2 and s.4 to about 200 X 200.

After moving and re sizing, your Splines should appear similar to the image at the right. Think of the Splines in a lofted object as "ribs".

Create a Loft NURBS Object, select all five Splines and drop them into the loft.



Be sure the Splines are in order. You should have a tapered form with a bulge in the middle and a cross-section (profile) of the Spline you first created. So far, you could have done this and more with a Sweep NURBS.

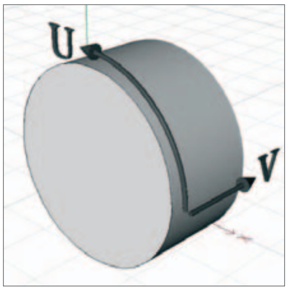
Be sure the Move Tool is selected and the Z-axis is locked. Select s.3 and activate the Point mode. From the perspective view (so you can see the results), experiment with the position and tangents of the points on s.3.

Now you are getting an idea of the versatility of lofting (the points for a loft Spline do not have to remain in the same plane but it gets a bit complicated when they are not).

Select the Model mode and rotate s.3 a bit around the P axis by selecting the Rotate Tool and dragging the P (red) band. Also, move the entire Spline around a bit to see the effects. Move the points on one or two of the other Splines, if you wish, to make a truly asymmetrical object.

You have much more control over lofted NURBS than the other types of Spline Generators. Different cross-sections allow more flexibility than can be achieved by a sweep rail. A very important point regarding lofted NURBS: All Splines should have the same number of points. This is one reason why we duplicated the first Spline to create the others.

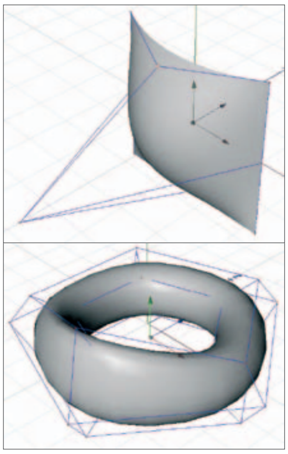
Another difference you will note from the Loft NURBS Object dialog is that Spline settings have no effect on the interpolation. Take a look at the loft's parameters in the Attribute Manager. The V subdivisions are along the length of the loft linking the Splines. The U subdivisions go around the object, as do the Splines. This way you do not have to change the interpolation of each of the individual Splines to change the subdivision around the object.



The Loft NURBS Object can make complicated objects, including faces. Although a lofted face would be difficult to animate. The trout pictured at the beginning of this tutorial is lofted. Its gills and mouth are animated by changing the shape of the Splines.

You can get rid of the need for caps by making the end Splines zero size. This makes for a pointy end, it takes several additional Splines to round it off and a lot of care to reduce any puckering that results from all the polygons coming together at the end point.

9. Bézier NURBS



We will finish up with a quick glance at various kinds of NURBS Object.

Fold and hide your Loft NURBS Object. Create a Bézier NURBS Object. In the perspective view, grab the point in the center and drag it along the Z-axis.

You may find it difficult to grab the point by trying to click on it with the *Move Tool* selected. This is because it is right on the axis' center. You can use the *Live Selection Tool* instead. Rotate the view enough to see the results. Imagine making a sail or a flag this way.

Undo any point moves and, in the Attribute Manager, change the Bézier NURBS Grid Points values both to 5 and check both Closed boxes. Play around with the points.

As you see, results are different. Bézier NURBS can fill certain needs as they are or serve as the base geometry for Polygon Models. Note the important differences to other NURBS Objects. There are no Splines used, the points generated have no tangents, nor can they be deleted to reshape the geometry and the surfaces do not pass through the points. It is as if the surface is a flexible plastic sheet suspended from the points by elastic cords, as with some tents. Imagine how powerful this would be if we could insert and delete points at will instead of being limited to the parameter settings. Better yet, what if we could use any polygon object with any geometry and have a function such as this generate a smooth surface which could be stretched by pulling on the points. All of this is, in fact, possible using the HyperNURBS Object. This is the most powerful modeling tool in CINEMA 4D and will be introduced in the next section after a discussion of polygon objects.

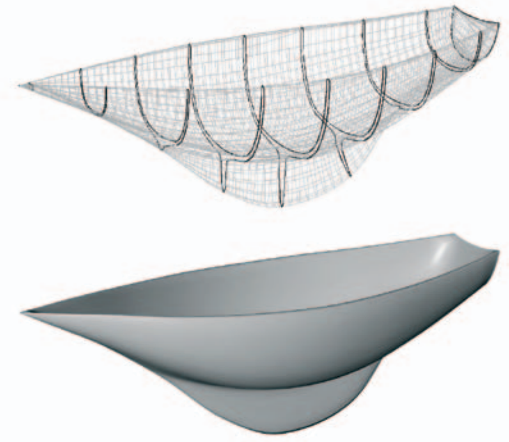


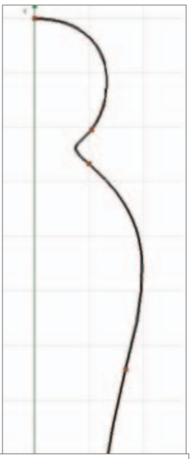
Table Lamp Tutorial

In this tutorial you will create a table lamp. We will continue past what is generally required for an object of this sort in order to practice more modeling techniques. Many of the lamp's components will be created using Lathe NURBS.

The lamp base

The majority of table lamp bases are radially symmetrical and suitably treated as Lathe NURBS. No surprise here, most ceramic bases are turned on a potter's wheel which is a "lathe for clay". If you want a figurine-type vase you will have to develop your polygonal modeling skills.

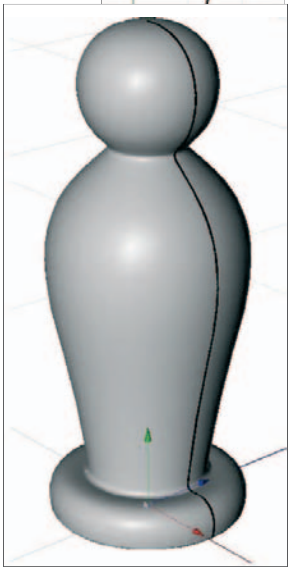
Create a new folder in your student folder titled "Table Lamp". Open CINEMA 4D and adjust the window. Set the focal length of the Editor Camera in the perspective view to 100. Save your new file to the "Table Lamp" folder as "Table Lamp 1". Save often, you will not be reminded again. Switch to the front view and create an open Spline similar to that at the right. Make sure the two end points are exactly on the Y-axis (X=0). Make your Spline about 100 units high (this is important so that the other parts will all be in proportion).



It is possible that you have experienced a little frustration because the axes for the last point created were exactly where you were trying to place the next point. In this case you end up moving the last point instead of adding another point. You could offset the new point a bit and tweak it later. Or you can choose another axis system.

Click on the Move Tool (the Rotate, Scale and any selection tool will also do). On the Modeling Axis Tab of the Attribute Manager, notice the Axis type is probably set to Selected. Change the type to Free and drag it out of the way. This only affects the modeling process, it does not change the object's axis.

Use the space bar to switch between the Live Selection tool and the Move tool to help you move the points around to get a nice look to your base. You may wish to change the modeling axis type back to Selected if you want the axis handy for moving along specific axes.



Name your Spline lampProfile and drop it into a LatheNURBS named Lamp Base. Use the Q key to toggle the NURBS on and off as you continue to tweak your spline.

The Spline pictured has nine points. Two are used for the lower right corner for a slightly rounded edge at the bottom edge of the base. It is important when modeling real world objects to avoid the artificial look of sharp computer-generated corners. Of course, how much effort you put into this also depends on how prominently the object will be featured in the scene. You may revise the Spline whenever you wish but it is important to keep all points and their handles in the ZY plane. This is best done working in the front view. You can work from the perspective view but then it is important to lock the Z-axis.

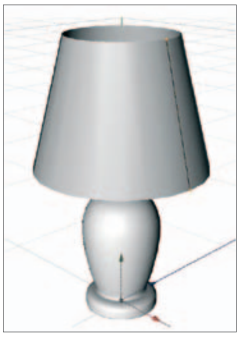
The lamp shade

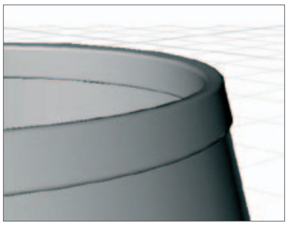
Now for a simple lamp shade with reinforced edges. To create the basic shape we could use a Cone primitive with a truncated (chopped off) top. However, since we are practicing with Splines and Generators, we will use another lathed object.

Create a two-point Linear Spline at about the position shown at the right. Name the Spline shadeProfile and drop it into a LatheNURBS named shade fabric. Shape the Spline (tweak mode) to get the look you want before proceeding.

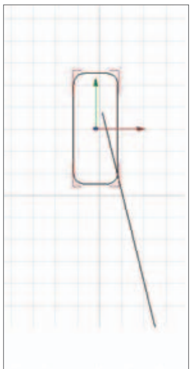
Although you can modify this Spline anytime, you are about to match other objects to it. These would also have to be modified if you further modify the fabric. Therefore it is best to finalize its shape now.

Take a look at your lamp in perspective view.



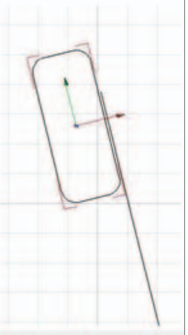


For many scenes you would be finished with modeling your lamp. Since the purpose of this tutorial is practice we will continue to add detail. Probably the most significant detail visible from most views is the addition of reinforced edges to our shade. These edges will not only add some detail and break up the otherwise smooth surface but will give the edges some thickness so they will not look paper thin if we move closer to the lamp. At the right is a detail view of the edge we intend to create. Try to anticipate the process as we proceed through the steps.



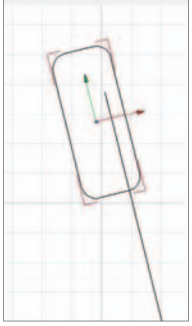
Hide the base and disable the lathe Deformer (shade fabric). In the front view, back the camera up until you can see the shade profile and the World origin. Create a rectangular Spline Primitive and scale it Width= 2, height= 5. Activate Rounding and set the value to 0.5. Name the Spline rimProfile and move it up near the top point of your shade spline. Enlarge the view until this rectangular Spline occupies most of the view.

We will now use the shade's profile to rotate the rim Spline until they are parallel.



Rotate rimProfile about its bank axis until its long edge is parallel to shadeProfile. Move rimProfile as necessary to bring one of its long edges close to shadeProfile to better judge when you have got the rotation matched.

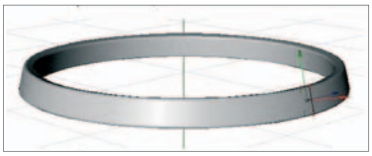
Again, a bit overkill for most applications. However, there will be plenty of situations where you will need this degree of positioning.



Now reposition rimProfile **to where the edge should appear.**

More of the edge has been positioned to the inside to suggest the fabric is wrapped from the outside onto the wire. This gives the shade rigidity. Now we are ready to lathe it.

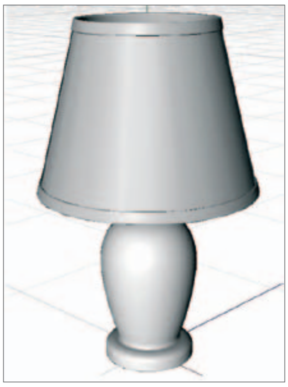
Drop rimProfile **into a new LatheNURBS named** top rim**.**



Since the bottom rim is the same except for its radius, we will copy top rim and adjust it.

Drag a copy of top rim and rename it "bottom rim". Working from the front view, drag bottom rim down along the Y-axis until properly aligned with the bottom of the shade.

It is time for a word about object origins. You may have noticed that CINEMA 4D, by default, places the origin of a newly created object at the World's origin. In many cases, this is fine but you may also want to reposition the origin for various reasons. In the re-positioning operation above, for example, you were probably moved in close enough to the shade to not see the World origin. Thus you could not directly grab the object's Y-axis to pull it down (in this case, it is probably less effort to lock the X and Z-axes than to pull the rim down. There is no real need here to reposition the axis). Sometimes, however, axis position is critical. This is especially true when objects are animated and modified using Deformers.



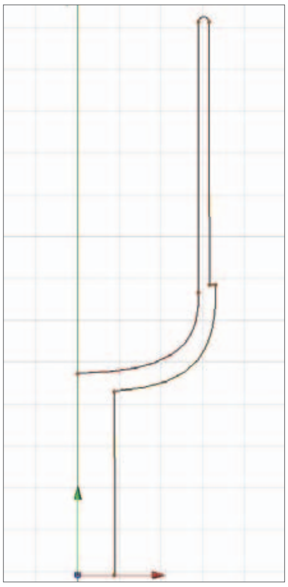
Next, you need to increase the radius of the bottom rim. Since this is a function of the position of the profile when it is lathed, you will have to move the profile out along the X-axis. However, remember, you rotated the profile Spline so its X-axis is no longer horizontal.

While the Move Tool is active, choose the World axis system. Drag the Spline rimProfile out along the X-axis until aligned.

A note on naming: You will notice we now have two objects with the same name - the Splines used as profiles for the two rims. If these were objects you might want to reference for various advanced features, such as targeting a camera to an object, you would want to be sure the names were unique. It is doubtful we will need to use these Splines in this way, so we will leave their names unchanged. Your lamp should now look similar to the one pictured above - certainly good enough for most applications.

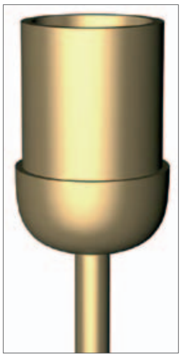
Drag the two rims and the shade fabric into a new Null Object named Shade**. Hide** Shade**.**

The socket



Let's model a socket. This object can also be easily created using a lathe.

From the front view, move in close and create a new Spline named socketProfile**. Use the Coordinates Manager to align the points to make the sides vertical.**



The Spline pictured has eight points and is about 13 units high. There is no need to close the lathe at the bottom, it will be hidden by the base. The long stem allows for adjustment, any excess will be hidden by the lamp base. If you want more realism you could use a real lamp socket and have it handy during the modeling process. We should also make sure the edges are slightly rounded to avoid the "computer-generated look".

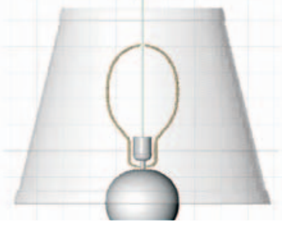
Since the object's origin affects operations such as scaling and how the camera view is rotated when the object is selected, we will reposition it.

Select the Object Axis Tool, grab the Y-Axis and move the object's origin up until it is located near the points of the Spline (above right). Drop the Spline into a *LatheNURBS* **named** Socket**. Switch to the perspective and do a test rendering.**

Now let's make the wires that support the lamp shade on the socket.

Working from the front view, create a Spline (wirePath**) that will serve as the path of a Sweep NURBS.**

Do not worry about the exact height, you will adjust it in a moment.



Create a *Circle* **with 0.3** *Radius* **for a profile, give it an appropriate name and create a** *SweepNURBS* **(**wire**) from your two Splines. For the wire on the other side, create a** *Symmetry* **Object named** shade support **and drop** wire **into it.**

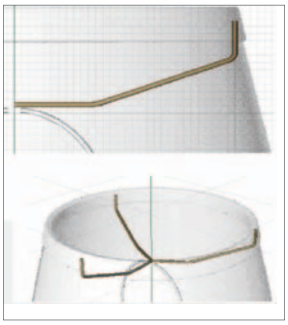
Hope you remembered that the Spline order matters for Sweep NURBS. The Symmetry Object's default setting is to mirror the object in the ZY plane (along the X-axis), which is perfect since we created our sweep in the front view. At the bottom of the wire, where it attaches to the base, we should have another component, probably fabricated from a Loft NURBS Object. Maybe we will have time for that later. First we need to adjust the wires for the placement of the lamp shade. We need to make the lamp shade transparent so we can work inside it but still see its shape.

Unhide and unfold the Shade **object. Select the** shade fabric**,** bottom rim **and** top rim **objects and check the** *X-Ray* **box. Adjust the points of** wirePath **until the proportions are as shown at the right.**

We now need a "yoke" to attach the lamp shade to its support.

Move close to the right top of the lamp shade and create a Spline (yokePath) to act as a path for the sweep shown at the right. Make yokePath a Child of a new SweepNURBS named yoke 1. Drag a copy of wireProfile into yoke 1.

Be sure the outer end of the yoke wire is positioned so it will disappear into the top rim of the lamp shade. We need two more wires to complete our yoke.

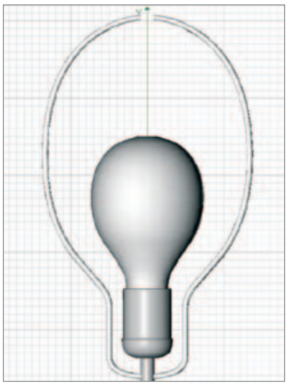


Drag two copies of yoke 1 and name them yoke 2 and yoke 3. Drop all into a new Null Object named shade yoke. From the top view, change the heading of yoke 2 to 120 and that of yoke 3 to 240.

Viewed from the top, it is a bit unnatural that one of the yoke wires should exactly line up with the support wires.

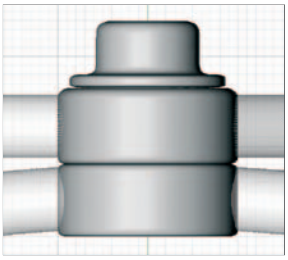
Rotate the heading of shade yoke an arbitrary amount. Make a bulb.

Shade and support connectors



Now you need a connector to join the support wires at the top, one to join the lamp shade yoke at the center and a nut to fasten the lamp shade yoke to the support connector.

Model as shown.



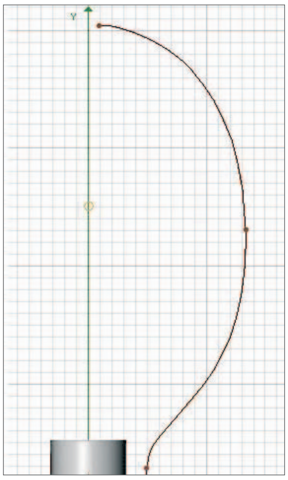
You can model the two connectors as one object but is best to create the nut separately so you can more easily apply a separate texture to it later.

Place all lamp components into a Null Object, name the Null Object Table Lamp.

Now you have a relatively detailed table lamp. If you wanted to use this in a scene you could copy and paste it or use the Merge command from the main File menu. In many cases, unless you have that special close-up on that mosquito planned, you would delete all components inside the lamp to keep the file at a reasonable size. For other close-up applications, you may want to increase the subdivision on some of the Lathe Objects for more smoothness (i.e., the base or the rims of the shade).

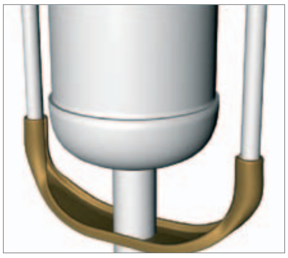
We will return to our lamp often. You will texture it in the materials chapter, turn it on in the lighting chapter and place it into a scene with other objects for final rendering. If you are working ahead of schedule, proceed with modeling the support connector that connects the lamp shade support to the base of the socket.

The support connector



The picture to the left will give you an idea of what the connector will look like. It is meant as a more realistic replacement of the bottom of the lamp shade support wires (if you are quite a bit behind the rest of the class you may be asked to call your lamp complete at this point).

Switch to the side view and move in on the base of the socket where the support wires currently attach. Apply X-Ray mode to the socket and shade support and hide any other visible objects. Create an open spline, as shown at the right, with its two end points at Z =0. Move the Spline's origin along the Y-Axis until it is in the Spline as shown in the illustration at the right.

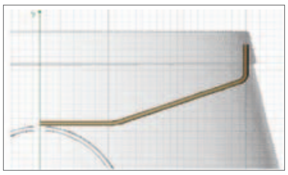


Get this as close as you can before proceeding. We are going to use the Mirror Tool to complete the other side.

Make sure your Spline is still selected, the Point mode is still active and no points are currently selected. Select the Mirror Tool (Mesh / Transform Tools / Mirror).

If any points are selected, the *Mirror Tool* will affect only these points and not the entire spline.

In the Attribute Manager, change Coordinate System to Object and make sure the Mirror Plane is the XY. Click on the Apply button in the Attribute Manager. Do not do any further operations - yet.



You should now see a symmetrical left half of your spline. We still need to do some work. The Spline currently consists of two groups of points even though they are one object. The two groups run in opposite directions. You can see this from the white-to-blue gradients traveling in both directions. We must reverse the direction of one of these groups before we can join them into one continuous spline. The reason you were instructed to do no further clicking is that the points of the freshly mirrored half are still selected. Let's reverse their sequence.

From the pop-up menu, choose Reverse Sequence.

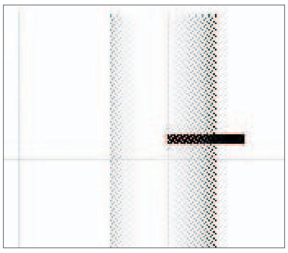
Now the points of both segments are lined up in the same direction.

Deselect all points and choose Join Segment (pop-up).

Now you should see just one color gradient from blue to white indicating that the points are all part of one segment. Two more steps.

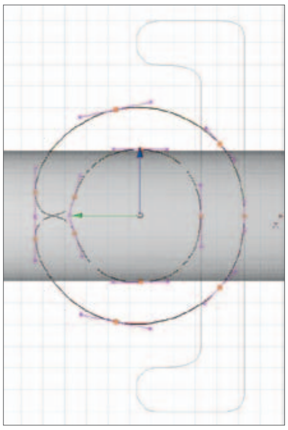
Make the Spline Closed (in the Attribute Manager).

Appearances are deceptive. Each of the two center points (on the Y-axis) actually consists of two points because of the mirroring operation.



From the main menu, choose Mesh / Commands / Optimize. Make sure the Points option is enabled selected before clicking OK (no points of the Spline should be selected or the operation will only apply to them and not the whole spline).

This will combine any points that are closer than the *Tolerance* value. Now your Spline is a single closed segment and will be easier to handle as we proceed with lofting. This was a lot of work but it ensures our Spline will not come apart when it or its Children are subsequently manipulated. Remember, lofting only produces predictable results if all the Splines have the same number of points arranged in the same order. So we will use this Spline to create the others in our loft object.



Name the Spline baseConnect and drop it into a Loft NURBS named base connector. Disable base connector for the time being. Copy baseConnect three times, keeping all copies at the same level of hierarchy within base connector. From the front view, arrange the Splines as shown at the right (their thickness has been exaggerated in the diagram so you can better see them).

Put your Splines in order so that the top Spline is at the top of the list and the Spline at the base (the first created) is at the bottom of the hierarchy. This is just so we can keep them straight in our discussion. Now we will modify the first (topmost) Spline so it will hold the wire in place.

From the top view, move in close to the top Spline.

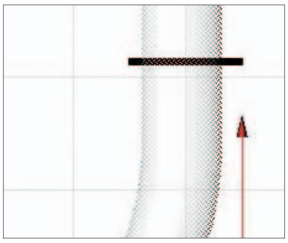
It would be nice if we could see the support wire in cross-section to best wrap our Spline around it. You could copy the support's profile and position it properly but you can get close enough just looking at the support wire running across the top in order to judge the thickness.

Move the points in a symmetrical manner until the Spline appears to wrap around the support wire.

The figure at the right shows the modified points selected, making their tangents visible (you could also delete half the Spline, modify it, mirror and join as you did above. Make sure the points begin in the same place and go around the same direction as the other Splines).

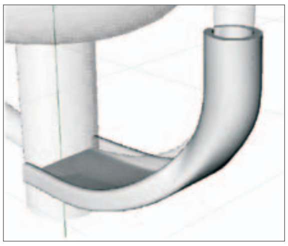
We want this cross-section to continue down the connector for a while to securely grab the support. To do so we will duplicate this modified spline.

Drag a copy of the modified top-most Spline so the copy is the second Spline in the list. Move it down in the Y direction only so it is under the top spline, as shown above. Activate the Loft NURBS Object.



At this point your loft should be looking somewhat like that to the right. The end looks too squared-off - we need to round it.

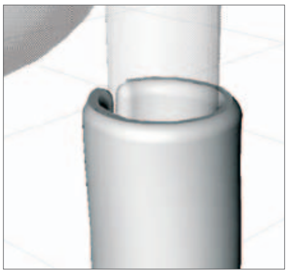
In the Caps Tab of the Loft NURBS Object, set Start to Fillet Cap with Steps set to 3, Radius set to 0.1 and check the Constrain box. Set End to None.



The *Constrain* option will ensure that the contour does not "swell" during rounding. The end of the connector shown at the right has had both U and V subdivisions doubled to appear smoother. Leave yours as is.

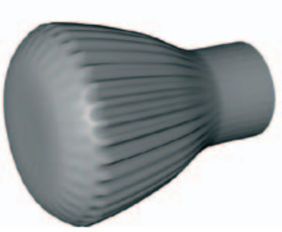
The last operation for the connector is to complete the other side. You also need to trim the support wire.

If necessary, adjust the height of the connector so its bottom rests on the top of the base. Edit the Spline used for the path of the support wire so it ends at the second Spline of the lofted connector and is straight while it is within the connector. Drop baseConnector into a Symmetry Object and admire your work.



You may notice, in the front view, that your connector appears a bit deformed at the bottom. This is because, when lofting, CINEMA 4D tries to make smooth curves through all the Splines. It does not know you would like the connector to lie flat in the center. This is easily remedied by slightly lowering the last Spline (at X = 0).

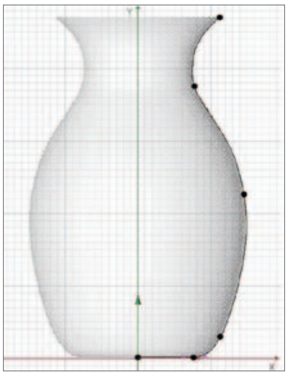
If you have still got time left, try modeling a knurled switch. The one to the left is a Loft NURBS Object which began with a star spline. A copy of the Spline was remodeled into a circle for some of the contours (how can this be done more quickly?)



Vase Tutorial

We will now create two vases. One will be a standard lathed object, the other will be fluted. You will also learn two new modeling techniques. Create Outline is a method used to add a parallel edge to a spline. Instances allow you to make copies of a Spline which can all be controlled as a single entity.

Lathed vase



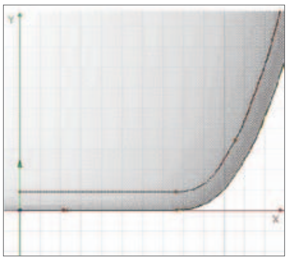
This is a pretty simple lathed object However, to model an even thickness for the walls, we will use a couple of new techniques. For an object as simple as this we could place points by approximation but the new techniques may come in handy for later applications.

Create a new folder in your student folder titled "Vases". Open CINEMA 4D and perform any usual setup adjustments (focal length = 100, etc). Save the new file as "Vases 1". From the front view, create an open Spline, as shown, using six points. Begin your Spline at the bottom center. Name the Spline vaseProfile and drop it into a LatheNURBS named Vase_ (lathed). Adjust your Spline until you are satisfied with your shape.

Although you could modify the shape later, it is best to get it right now. After you have created the inner walls any further modeling will require extra care to keep the inside and outside surfaces parallel.

While still in points mode, and with your Spline still selected, select Create Outline from the pop-up menu. Drag a little to the left to make the inner wall of the vase.

The procedure has left us with a little cleaning up to do. It has closed our Spline (we want it open where it swings around the Y-axis) and the top edge has been flattened.



Open the Spline via the Attribute Manager (uncheck Close Spline).

If you were careful to start your Spline as instructed, the opening should be in the proper place at the bottom.

Move in for a nice close-up look at the top edge. Select the two points that make up the edge. From the pop-up menu, select *Equal Tangent Length*.

Now you have some handles with which you can adjust the curvature of the edge.

While holding down the Shift key, drag each of the two new handles (keeping each in line with its opposite until each is about half the thickness of the edge.

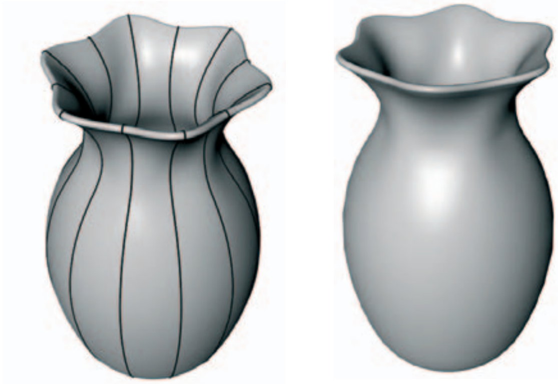
Not using the shift key would have kept the two sides of the tangent handles in line, but moving the top handle would have reoriented the other side, disturbing the parallel walls. It will probably take you longer to texture a vase than to make one once you get adept at quickly creating smooth Splines.

For many purposes, the parallel walls will not matter much. However, this vase can be textured with a glass-like material and will still look good when transparent.

So Lathe Objects are probably getting pretty easy to create by now. However, how would you make a fluted vase like that to the right? In real life, this would take more work because it could not just be turned on a wheel. Likewise, our 3D modeling process will take longer and show us a few new concepts and tools such as Instances. This vase is based on the same Spline as our first vase but we will create several copies and loft them instead of lathing.

The fluted vase

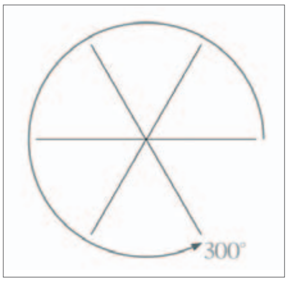
The figure below shows the basic plan of attack. We will use 12 Splines and rotate each 1/12 of a circle around the Y-axis. Drop these all into a *Loft NURBS* and it will do its magic, stretching a surface around the Splines as if they were ribs. We really only need two types of spline, ones with narrow tops and ones with wide tops. If we alternate them going around the circumference, the loft will swing in and out to accommodate them and give us our flutes.



Drag a copy of vaseProfile out of vase 1 and hide vase 1. Name the copy fluteProf_A. Drag a copy of fluteProf_A and name it fluteProf_B.

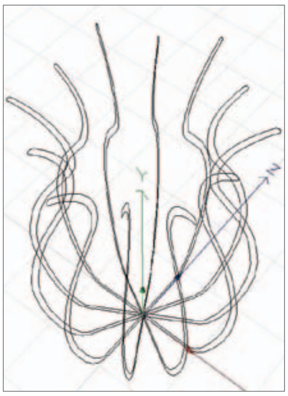
These are the two basic Splines we will alternate to create our flutes. We will rotate fluteProf_B and modify its points later. First, we will create "instances" of these two Splines which will save memory and time. More importantly, we can modify the original Splines at any time and all the instances will assume the changes as well.

Select fluteProf_A and, from the main menu, the Tools / Arrange Objects Duplicate function. In the Attribute Manager, make sure all tabs are active. Set Copies to 5 and select the Instances option in the Clone Mode menu. Change the Mode to Linear. Disable the Enable box under Position; check the Enable Rotation box under Rotation. Enter 300 in the first text box after Rotate (heading) and click Apply.



The heading value of 300 means the last copy will be placed at 300° counterclockwise around the Y-Axis (heading) from the original, which is at 0 degrees. Now we have five new Splines rotated around 300°, each Spline is 60° from the last (300°/5 Splines = 60° per spline). Another 60° from the last to the first will complete the circle. If you unfold the new Null Object containing the five instances you will note that numbers have been appended to their names, in the order in which they turn counterclockwise around the circle.

Do exactly the same duplication for fluteProf_B.



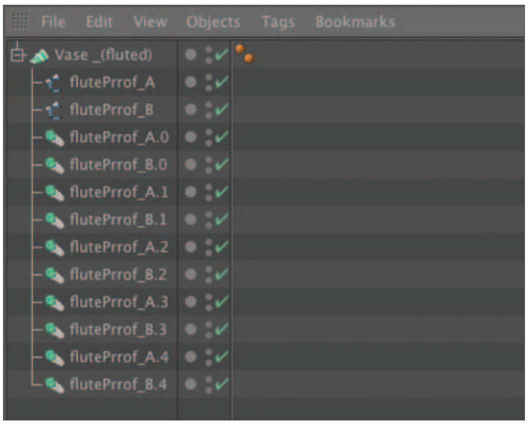
This group of Splines need to take up the spaces between the Splines of fluteProf_B.

Select the Null Object containing the instances of fluteProf_B and the original fluteProf_B. Rotate their headings +30°. Check the orientation in the top view.

Our 12 Splines should now all be equally positioned every 30° around the Y-axis. Now we must make sure they are in the correct order within the loft object.

Create a Loft NURBS, name it Vase_(fluted) and drop your Splines in order from the very first (fluteProfA) counterclockwise around the Y-Axis. Delete the two Null Objects (which should now be empty).

The order should be: fluteProf_A, fluteProf_B, fluteProf_A.0, fluteProf_B.0, fluteProf_A.1, fluteProf_B.1, etc. You can probably see that there is no lofting between the last and first Splines.



Vase _(fluted)
flutePrrof_A
flutePrrof_B
flutePrrof_A.0
flutePrrof_B.0
flutePrrof_A.1
flutePrrof_B.1
flutePrrof_A.2
flutePrrof_B.2
flutePrrof_A.3
flutePrrof_B.3
flutePrrof_A.4
flutePrrof_B.4

Check the Loop option (in the Object Tab) for the loft object. Render the vase in the perspective viewport.

Your vase probably looks a bit chunky along the vertical direction.

Change the Mesh Subdivision U to a value of 60 and render again.

The U direction is along the Splines (top to bottom in this case). You may need to increase the subdivision, depending on how the vase will be used in the scene. This vase should look almost exactly like the lathed vase. The difference between the two:

While in the perspective view, choose the Point mode and select fluteProf_A. Activate X-Ray mode for the loft. Select the two top (rim) points of fluteProf_A. Choose the Move Tool. Drag the X-axis and observe the results.

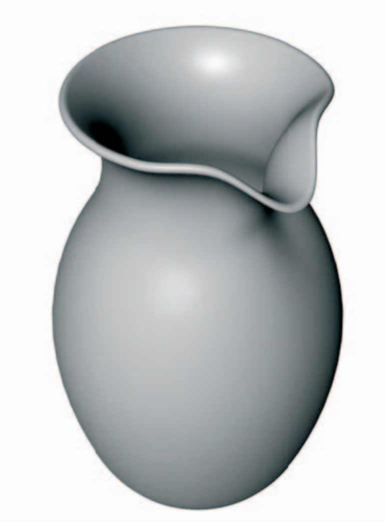
All of the instances, every other Spline in this case, respond in kind to movements of the master object.

Adjust the flutes to a desirable number. Also, move the next pair of points down and out a bit to carry the transition farther down the vase.

You may feel the top is a bit large in diameter. You can also manipulate the shape of the Splines alternately by moving points on fluteProf_B. Make sure to always select the points in pairs, both inside wall and outside wall. You may also add some movement along the Y-axis but not along the Z-axis.

You should, based on what you have just done, be able to make a pitcher with a pouring spout like the one at the left. If you have time remaining, make this pitcher.

You could improve on the pitcher by using several different Splines for contours. We will later investigate other, more powerful ways of adding variation to this form which include the ability to add seamless handles.



Splines and Generators

In this exercise you will model objects using Splines and Generators. The first group of objects are simple enough to require only one Generator. The second set will require a combination of objects in addition to Generators. You will be given various amounts of help and hints depending on whether new techniques are required.

Open and adjust CINEMA 4D. Save the file as "Splines Practice" to the "Intro Tutorials" folder. Create two new Null Objects. Name the bottom one "objects" and hide it. For the name of the second Null Object, enter a single space.

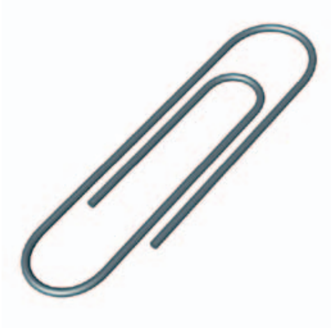
The objects Null Object is to put your objects onto as you finish each to keep the manager uncluttered. The other Null Object serves as a spacer to help visually place the inactive objects away from those You are working on. Both are strategies to keep the work area tidy. As you create objects, name them as they are named in the exercise and drag them into the objects Null Object.

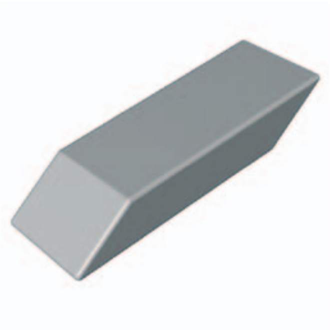
The most important step is to figure out which Generator or Generators to use before investing time in actual modeling.

One generator objects

Paperclip

The paperclip should be easy to model. The main consideration is to construct the path carefully; make frequent use of the axes (or the axis locks) when moving points and handles to keep your sides straight and parallel.



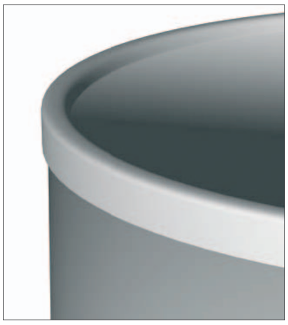


Eraser

You can either start with a rectangular primitive spline, make it editable, move the top two points over and *Chamfer* the corners. Or you can build the Spline from scratch.

Tin can

Keep the sides straight, add a very slight curvature to the top and bottom (not too much). The only question is, are you going to model the rim twice for both top and bottom?



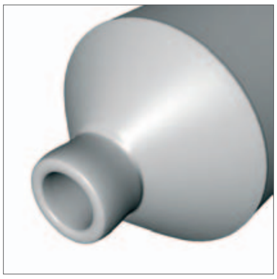
Cookie cutter

The *Create Outline* functions works on closed Splines also. Anytime a Spline is composed of two unconnected, closed segments, one contained within the other, the inside segment is used to create a hole. Make sure either all or no points are selected when using *Create Outline*.

"My name"

Use your name or the name of the class. Be conservative with the rounding or the text will "balloon". Try a couple of different fonts. Later we will complete an entire project using text for logos.

Paste tube



Nine contours were used for this Tube. More may be used if you want more realism (although polygonal modeling should then be used). To get sharply defined transitions such as at the point the tube begins to narrow toward the hole, place two identical contours close together. You could model this further to create the hole at the end (left) with two more Splines. Complete this step if you are working ahead of the class (and, using another two Splines, you can make the hole's edges sharp).

Picture frame

A rectangular primitive, with some of its parameters adjusted, is one of the objects used in the picture frame. To get the other Spline oriented properly after constructing it you may have to rotate it around its pitch axis using the *Object Axis Tool*. Remember, the view in which you create the profile is very important.

Multi-generator objects

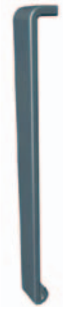
Mechanical pencil

This pencil (far right) could be created with two objects, but it is best to use five so that it can be more easily textured. The clip is shown in detail at the near right.

Desk lamp

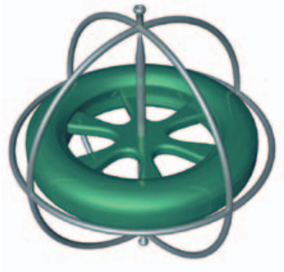
This object also uses five simpler objects to make up its shape. Remember to also use a little Rounding on the bottom of the base as well. A close up of the joint between the neck and the base is also shown.

Model the shade around the Y-axis, like you have done with all lathed objects, then move it into position later.

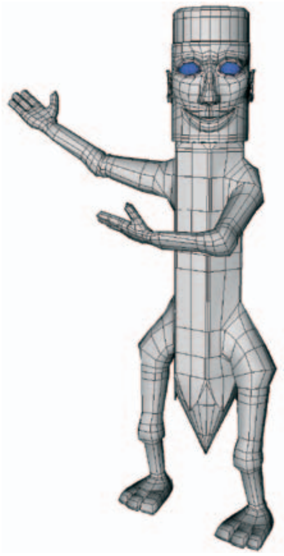


Gyroscope

Here we will use eight sub-objects and three different Generators. Three of the objects are simply other objects that have been duplicated and moved. This will be easy if you keep all object origins in the center. The webbing holding the mass in place uses one of the Spline Primitives.

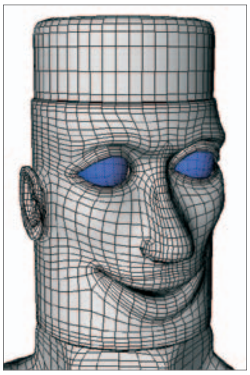


Polygon Modeling



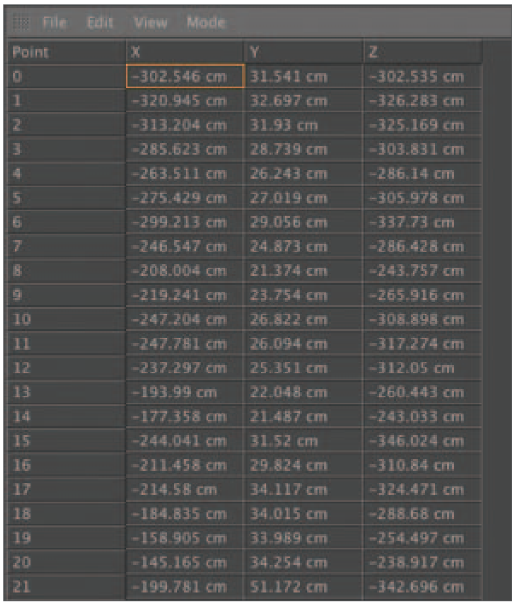
Overview

Parametric objects and objects formed from applying Generators to Splines have many applications in 3D modeling. However, complex objects requiring fine modeling or detailed control of surfaces during animation are generally created with polygonal modeling. Modeling at the polygon or point level gives you the ultimate in control. One drawback of polygonal modeling is the slowness and, often, tediousness of working point by point. However, there are many point and Polygon Modeling tools to help speed up the process. The product of polygonal modeling is an object made entirely of polygons. To reduce the faceted look of a polygonal surface which would necessitate enormous numbers of polygons, CINEMA 4D has a special Generator, the HyperNURBS Object. Since this versatile and valuable tool is more often than not used to smooth polygon objects, it is discussed later in this section. We will end our discussion of polygonal modeling with a look at deformers which act on polygon objects and parametric objects alike. These deformers include Bend, Twist, Wrap and Explode, among others. Deformers can be used in animation as well as modeling. Some special deformers such as bones are used in more advanced modeling techniques such as character animation.



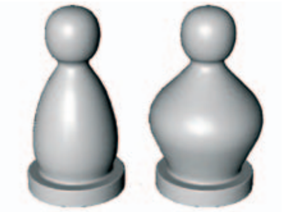
Polygons and points

As mentioned previously, all objects are ultimately rendered as a set of polygons defined by a collection of points. For polygon objects, the computer maintains a list of these points and how they should be joined to form connected polygons. You can view and edit this list using the *Structure Manager*. Primitives and Spline objects calculate these lists on-the-fly when it is time to render, based on the parameters you have set.

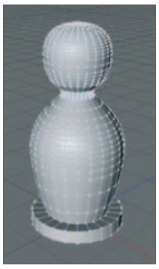


Point	X	Y	Z
0	-302.546 cm	31.541 cm	-302.535 cm
1	-320.945 cm	32.697 cm	-326.283 cm
2	-313.204 cm	31.93 cm	-325.169 cm
3	-285.623 cm	28.739 cm	-303.831 cm
4	-263.511 cm	26.243 cm	-286.14 cm
5	-275.429 cm	27.019 cm	-305.978 cm
6	-299.213 cm	29.056 cm	-337.73 cm
7	-246.547 cm	24.873 cm	-286.428 cm
8	-208.004 cm	21.374 cm	-243.757 cm
9	-219.241 cm	23.754 cm	-265.916 cm
10	-247.204 cm	26.822 cm	-308.898 cm
11	-247.781 cm	26.094 cm	-317.274 cm
12	-237.297 cm	25.351 cm	-312.05 cm
13	-193.99 cm	22.048 cm	-260.443 cm
14	-177.358 cm	21.487 cm	-243.033 cm
15	-244.041 cm	31.52 cm	-346.024 cm
16	-211.458 cm	29.824 cm	-310.84 cm
17	-214.58 cm	34.117 cm	-324.471 cm
18	-184.835 cm	34.015 cm	-288.68 cm
19	-158.905 cm	33.989 cm	-254.497 cm
20	-145.165 cm	34.254 cm	-238.917 cm
21	-199.781 cm	51.172 cm	-342.696 cm

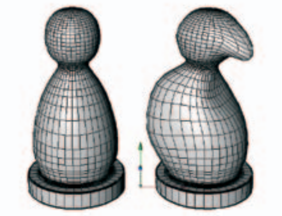
Although you could create a polygon object from scratch, it will invariably begin as a primitive. The primitive is then modified in various ways that affect its geometry and how it is subdivided. Deformers may also be applied to help shape the object which is subsequently converted into a polygonal object with the *Make Editable* tool. Beyond the usual undo process, this is not reversible. The object is henceforth a polygon object.



When a primitive or NURBS Object is converted to a polygonal object you trade the ability to easily make changes to the overall shape (right top) for the fine control over small regions of your object (right bottom). For example, you have lost the ability to make your pawn fatter just by dragging one point on a spline. You could still make the polygonal version fatter but by more laborious means, such as selecting and scaling groups of points. However, with the polygonal version, you can make asymmetrical changes such as bulges (a "nose" for your pawn) which cannot be done while the object remains a Lathe NURBS.



In addition to the fine control needed to model irregular and organic



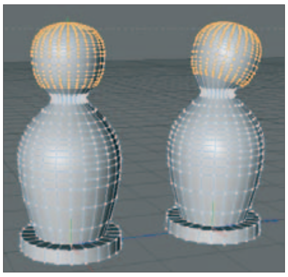
objects, Polygon Models offer much more control during animation. PLA, or Point Level Animation, allows individual points to be animated allowing changes of shape. Also, a group of points may be assigned to a bone and will move in response to the animated movement of the bone. Techniques such as these allow eyes to wink and character's mouths to speak.

Once an object has been made editable, you may manipulate either its polygons, points or edges of polygons using the Polygon mode, *Point* mode or Edge mode . Since the corners of the polygons are edges and these are determined by points, moving one affects the others. Some operations are easier using points, others are easier with polygons or edges.

Selecting Points/Polygons

The Live Selection, Rectangle and Free Selection Tools all work with points and polygons. The *Shift* and *Control* keys work as modifiers in a similar way as for Splines. The Select menu lets you modify your selection in many ways. This is indispensable for complicated objects. For example, it can be quite tedious to select only the points you need in order to adjust the mouth on a head containing hundreds of points. The selection tools allow you to selectively hide points and polygons to make this much easier. The *Set Selection* menu item lets you save sets of points which can later be used for quick access to only those points. This is handy for many purposes in modeling or for actions such as movement by a bone or restricting the placement of a texture. The Set Vertex Weight menu item lets you assign different weights to points in a set which determines what degree they will be affected by certain actions such as movement of bones.

Moving Points/Polygons

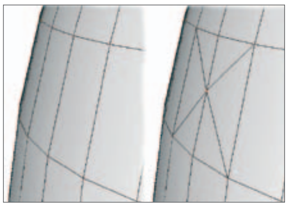


The Move, Rotate and Scale Tools all work with points, edges and polygons. Note that rotate and scale require more than one point be selected. The operation is then performed as if the selected points were an object of their own. The Coordinates Manager can also be used with points. In the case of a single point the position values are for that point. Otherwise they specify the center of a cube containing the selected points. You may also edit a point's position in the Structure Manager although this is not generally necessary.

Point/Polygon modes

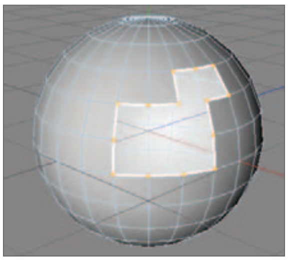
When modeling polygonal objects, you must work directly on the object's geometric structure. This structure is also referred to as the **mesh** or the **cage**. Under Mesh in the main menu bar are numerous tools devoted to manipulating the mesh's points, edges and polygons. Some of the essential tools will be discussed here, others will be introduced in the tutorials. Some are quite specialized for advanced modeling.

Create **point**



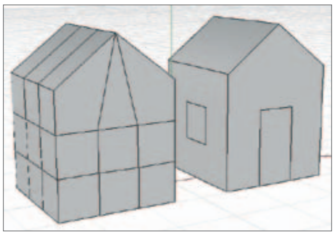
As the name says, this tool creates a new point in the plane of the polygon under the position of the cursor. If you click while an edge is active, the point will be placed on the edge. One disadvantage of adding points in this manner is that triangular polygons will be created. While for some objects this does not matter, quadrangles (four-sided polygons) are generally better for Hyper-NURBS Objects. Other tools and techniques will allow you to better maintain "quads" in your object.

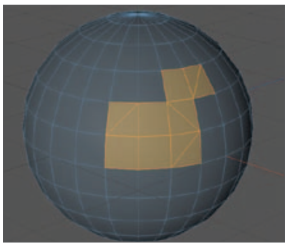
Close polygon hole



All three of these tools can add polygons to your object. Close Polygon Hole will fill a hole in the mesh of a polygonal object by adding a single polygon. The new polygon will have as many edge segments as the hole's perimeter (image at right). If the hole has more than four edges, the polygon created is termed an N-gon, which means it has more than four edges (line segments between points). N-gons are useful for architectural modeling and situations where you would otherwise have to create a lot of polygonal grid work just to add detail to a local area. Use of N-gons can make these types of objects less cluttered. However, N-gons are still divided internally by CINEMA 4D into quads (four-sided polygons) and triangles (actually, all polygons are reduced to triangles during rendering). At the right is an example of how N-gons can simplify the look of an object, at least in the editor.

Create Polygon

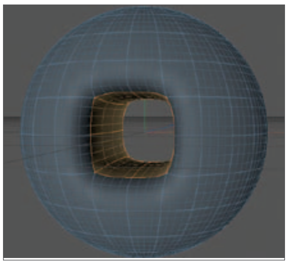




The *Create Polygon tool* also fills holes in a polygonal object but offers more control. Whereas *Close Polygon Hole* automatically fills in the entire hole in the mesh with a single poly, the *Create Polygon tool* lets you control the placement and number of edges (sides) of each poly. This lets you connect three or four points to make polygons. Simply click on three or more points sequentially, ending with a double-click on the first point. You can even add additional points and polygons outside the boundaries of the existing mesh.

Although you can create a polygonal mesh from scratch with this tool, it is usually used to remodel existing polygonal objects. The tool is also useful in connecting two meshes which were easier to model individually, such as the arms, torso and head of a human character.

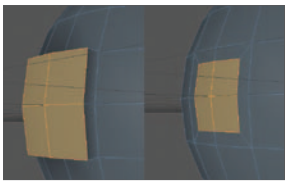
Bridge



Bridge can be used in point mode much like the *Create Polygon tool*. Click on one point and drag to the next to create one side of a polygon. Do the same for the other side and there is your poly. In edge mode, drag from one edge to another to create polygons.

Bridge has another function in Polygon Mode. It has the effect of taking two selected polygons from different areas on an object and connecting them with new surfaces (right). This tool is also useful in connecting separate surfaces, such as arms to a torso.

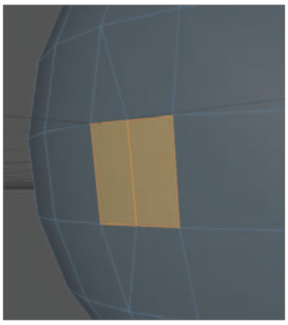
Extrude and extrude inner



These are possibly the two most important tools for creating polygonal models. Extrude will take a selected polygon or group of polygons and extend them outward or inward, perpendicular to the surface at that point. New polygons will be created to keep the extruded polygons connected to the object's surface. A major advantage over other methods of adding polygons is that the quadrangular nature of your model is retained.

The Extrude Inner Tool works much the same except that the selected polygons are shrunk within the same plane. The Attribute Manager offers control over some parameters associated with these tools.

Knife



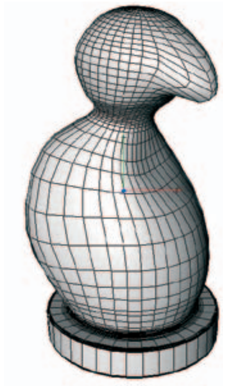
The *Knife* Tool is used in Polygon Mode to subdivide a selected region. Everywhere the tool slices across a polygon edge a new point will be created. You can restrict the tool's action to selected polygons, otherwise the slice will affect the entire object. This is a useful tool but must be used with care. Because polygons are limited to no more than four points, triangles will be created at the ends of your selection unless it spans the entire perimeter of the object. Study the difference between using the knife tool (to the right) and the Inner Extrude Tool pictured previously.

Subdivide

As the name implies, this tool subdivides the selected polygons a specified number of times. If the HyperNURBS Subdivide box is enabled, the curved surface of your HyperNURBS Object will be preserved. Triangles will be created where the subdivided area meets the area which was not selected.

Optimize

The *Optimize* tool gets rid of the extra points and polygons that are sometimes left after a long, complex modeling session. The most useful function is to consolidate points at seams where you have joined objects. This might happen, for instance, when joining an arm to a torso or when joining mirrored objects.



Magnet

The Magnet tool exerts an attraction on points within a certain radius from the cursor. The radius of influence as well as the shape of the magnet's influence can be controlled in the Attribute Manager. The magnet acts only on selected points unless none are selected in which case it acts on the entire object.

Normals

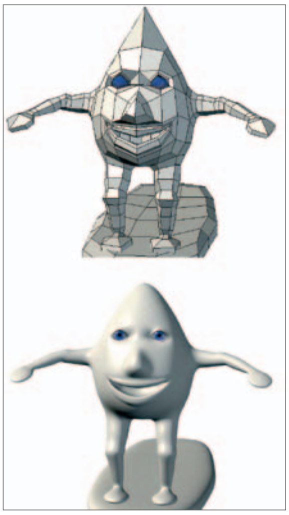
Every polygon has a front face and a back face. At the center of each polygon you may see a line perpendicular to the plane of the polygon and facing toward the front of the polygon. This is called the normal (normal, in mathematical usage, means perpendicular).

Normals affect how texturing is applied to surfaces. Sometimes, when creating new polygons in an object, normals will point in the opposite direction of the others as a result of how you created the polygons. When this happens, you should either use *Reverse Normals* on the offending polygons or select all polygons and use *Align Normals* .

There are also functions such as *Normal Move* which moves a polygon along the axis of its normal. Normals also affect which faces disappear when you select *Backface Culling* from the Options menu to avoid looking at the back faces of your objects.

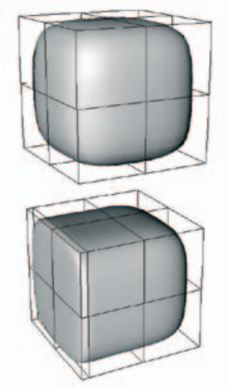
All of these tools will become more familiar as you use them in tutorials and projects of your own. You should also freely refer to the Help files for more detailed information and suggestions for usage.

HyperNURBS objects



Undoubtedly, HyperNURBS Objects are the most powerful modeling tool in CINEMA 4D. They allow you to create smoothly curved surfaces where the rendered polygon count is high but the number of polygons actually used to model may be quite low. This makes it much easier to model large areas of surface with moving of just a few points.

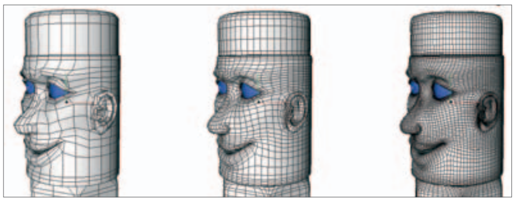
When a polygon object is dropped into a HyperNURBS Object it turns into a "cage". The surface of a polygon is now suspended from the points at the corners of the cage, as if hung by elastic bands. Depending on the proximity of adjacent points, the resulting surface will have a certain amount of curvature.



The curvature at a certain position on the surface of a HyperNURBS is determined by the density of points above that region. Where points are more concentrated, the surface will be more angular, as if the "elastic bands" were pulled tighter and the surface pulled closer towards those clustered points. A method of obtaining a sharp or angular edge with HyperNURBS is to move a selection of points over an adjacent set until their positions coincide.

Successful modeling with HyperNURBS depends on a few strategies and techniques. First, you should keep the polygon count of your mesh as low as possible. This will result in a smoother surface since the subdivisions used for rendering are not directly based on your mesh but on the *Subdivision* settings in the Attribute Manager. A low polygon count is also easier to modify and results in faster renders and redraws.

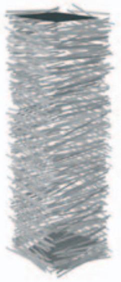
Second, as mentioned before, try to restrict your polygons to quadrangles if possible. This creates a smoother surface with less chance of artifacts appearing during rendering. In practice, this means maximizing use of the *Extrude* and *Extrude Inner* Tools and minimizing use of the knife tool.

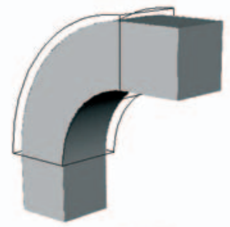


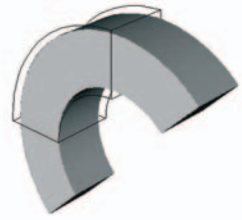
HyperNURBS Objects offer an efficient, effective way of modeling all sorts of curved and organic shapes. Although you can model humans and other animals as Loft NURBS, they are almost always done by applying a HyperNURBS to a polygonal mesh. You will learn more about HyperNURBS as we proceed through the tutorials. Pencilman's head was modeled using the mesh below left. The other two figures show the effects of applying a HyperNURBS to more finely divide and smooth the mesh for rendering.

Deformers

Deformers are used to modify existing objects either during modeling or during animation. The Deformers include Bend, Twist, Bulge, taper, Wind, Wrap, Explode, Melt and Shatter. Another Deformer, Formula, lets you enter mathematical formulae which deform the object according to parameters you define. Two other special deformers, bones and FFD (Free-Form Deformation) are used for advanced techniques.

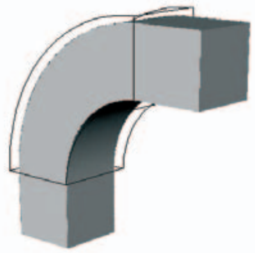


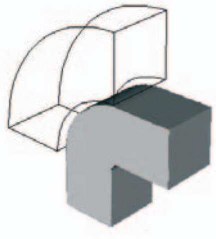




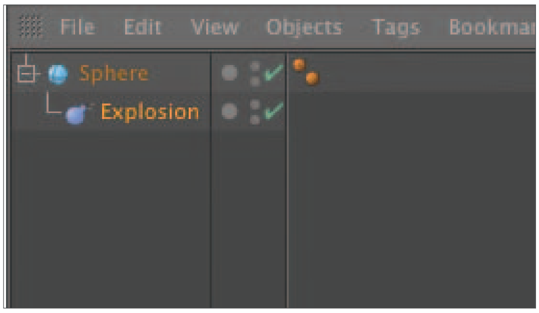
The Bend, Twist, Bulge and Taper deformers have similar input boxes that allow you to adjust the size of the Deformer and the extent of its influence. With the *Bend* Deformer, for example, the amount of bend is defined by the *Strength* parameter. You will also notice an orange handle in the editor window allowing interactive control of the deformation. The orientation of the Deformer is also important. The deformation always occurs along the Y-axis of the Deformer. You may need to rotate the Deformer to achieve the desired result.

Each of the above deformers has a *Mode* parameter. This limits the influence of the Deformer on the object in various ways. *Within Box* (upper left) will cause the deformation to only be applied within the Deformer's bounding box. Any part of the object outside this boundary will remain in its original position. The *Limited* setting also restricts the effects to the Deformer's bounding box but the ends of the object outside the box will be moved so that they extend naturally from the ends of the deformed portion. The *Unlimited* setting causes the deformation to be applied to the entire object, regardless of the relative size of the Deformer's bounding box, as compared to the object.



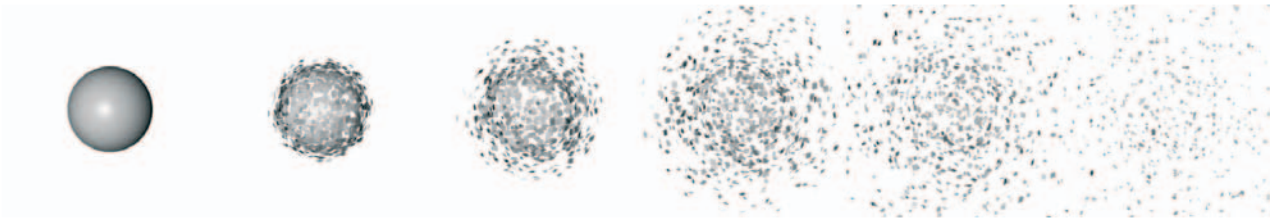


Different effects can be achieved depending on the position of the Deformer relative to the object. This can be used in animation. For example, an object can be made to fluidly turn a corner by moving a 90° Bend Deformer down its length. The best way to learn more about how Deformers work is by using them. The Help files also offer more detailed information about using deformers.

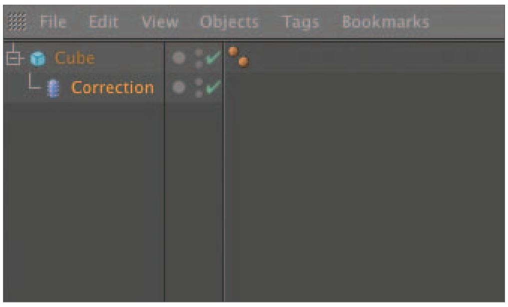


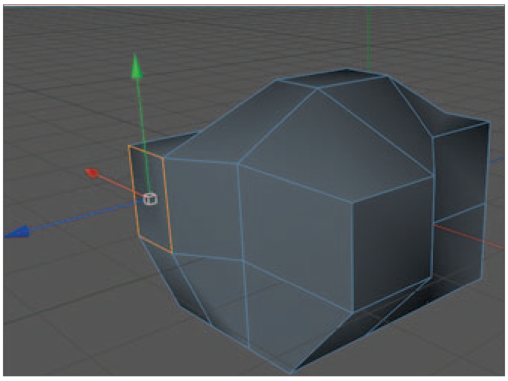
The *Explosion*, *Melt* and *Shatter* Deformers are generally used in animation only. Each has a *Strength* parameter which dictates to what extent the object should be exploded, melted or shattered.

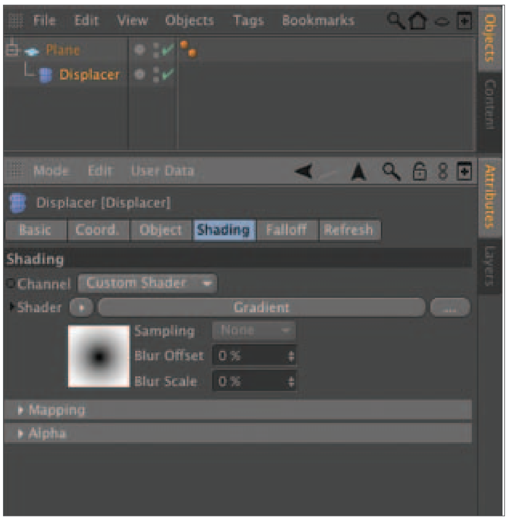
To use Deformers, simply drop one into the object you wish to deform. They work with Polygon Objects, NURBS Objects and Primitives. Their effects are dependent upon the amount of subdivision of the Parent Object. Smoother bends will require more subdivisions along the relevant axis. Several deformers may be used with the same object. If this is done, their order in the hierarchy makes is important, e.g. a bend placed before a twist will yield different results from a twist placed before a bend.



With the release of CINEMA 4D 12 many new deformers were added to the software and we now have some new ways to use them. For example the Correction deformer, when placed as a child of any primitive object, will allow you to edit the position of its parametric polygons, edges and points. You can, for example, create a parametric cube and with the addition of a Correction deformer you can move individual polygons to create a completely different shape. At any time you can disable the Correction deformer to see the shape snap back to the original cube. This gives you similar modeling capabilities to those you would have with editable objects while still allowing you to alter the overall size, shape and polygon density of the object through its parameters. Another recently added deformer is the Displacer deformer. With this deformer you can alter the position of the points making up a surface using the luminance values from a material. A simple example would be to place the Displacer deformer as a child of a parametric plane. If a Circular gradient is chosen for the Shading parameter the result is a bulge along the Y axis at the centre of the Plane. A more interesting effect can be achieved if Noise is used instead of a gradient. This provides a particularly useful technique for modelling liquid surfaces as the noise can be animated. Depending upon the type of noise used effects from turbulent rapids to majestic ocean swells can be achieved relatively quickly.







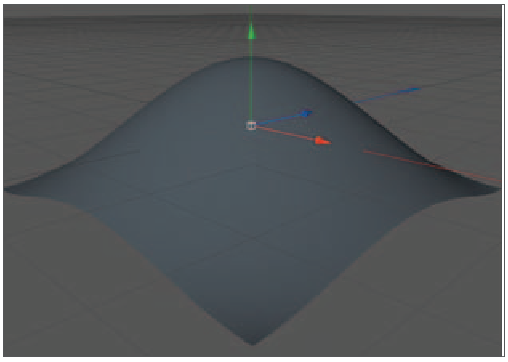
Channel Custom Shader

Shader Gradient

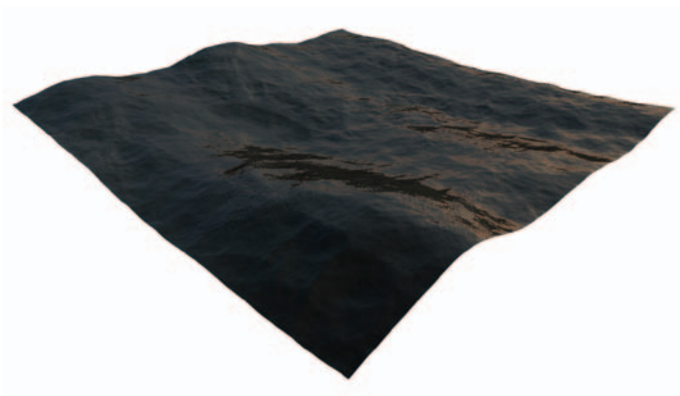
Sampling None

Blur Offset 0 %

Blur Scale 0 %



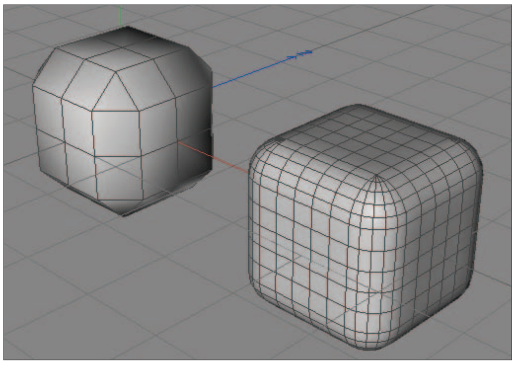
The modeling methods discussed here will allow you to create complex and sophisticated objects, especially those suitable for animating. A polygonal mesh inside of a HyperNURBS Object will offer maximum modeling and animating flexibility, while producing a smooth surface with the lowest polygon count. These objects can be modified further with the help of deformers or point level animation. Most of your complex objects, especially those designed for character animation, will be done at the polygon/point level. This offers the most flexibility in both the modeling phase and during animation. However, working at this level is time-consuming and requires patience. Most importantly, successful polygonal modeling requires good planning and technique to get the results you want in a reasonable amount of time. This is an introductory tutorial to demonstrate certain tools and basic techniques. You will gain more insight as we work through further tutorials devoted to modeling specific objects. This is a good time to begin to study the work of others, paying attention to the underlying geometry (the polygonal mesh) which they have constructed to create their objects.



Polygon Modeling Tutorial

1. Editable objects

Although polygonal meshes, also called cages, can be created from scratch, you would need a very special reason to do so. The general process is to begin with a primitive or, less often, a Spline NURBS Object, and make it editable. The *Make Editable* command converts your selected object into a polygonal object whose polygon count is directly related to the subdivision parameters you have previously set for the object.



Open a new CINEMA 4D file and adjust as usual. Save the file in the "Intro Tutorials" folder as "Polygon Tutorial". Back the camera away until you see all World axes. Create two cubes and name one Cube 1 and the other Cube 2. Move Cube 2 to an X position of 400. Rotate and translate the view until you can see the two cubes mostly from the front but still having some of the top and side visible. Set the parameters of Cube_1 to 2 Segments for the X, Y, and Z directions, check the Fillet box and set Fillet Radius = 40 and Fillet Subdivision = 1. For Cube_2, set all Segments values to 5. Check its Fillet box but leave the fillet parameters at their default values of 40 and 5. Switch to the Gouraud Shading (lines) display mode.

Nothing surprising here, you have seen the results of segmentation before.

Eliminate the Phong Tags. Choose the Polygon mode.

With the smoothing (Phong) function eliminated you can see how the surfaces will be broken up into polygons for rendering. However, these are not yet polygonal objects, as evidenced by the fact that you can still easily modify parameters in the object's dialog.

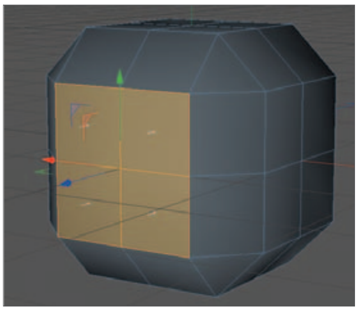
Convert the objects into polygon objects by selecting both and clicking on Make Editable in the Command Palette, selecting Make Editable from the main Mesh / Conversion menu or using the keyboard shortcut C. Select Cube_2.

Now, on the selected cube, you see the light-blue lines that represent the boundaries of each polygon. (in these tutorials, polygon borders have been made darker to make them more visible). Notice that polygon icons have replaced the Cube Primitive icons in the Object Manager.

Select one of the polygons and inspect the Attribute Manager.

All you can "adjust" are the coordinates and the basics such as name and visibility. Now you have a true polygon object: No parameters to adjust. You must now manipulate these objects via their points or polygons. This can be done by directly manipulating points/polygons in the editor or with the help of deformers.

2. Moving polygons and points



As you progress through this section, remember that almost any operation applicable to moving polygons also applies to points.

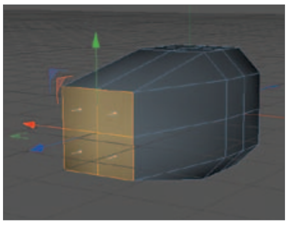
Move in for a little closer look at Cube 1. Using the Live Selection Tool, select the four polygons on the front face of Cube 1. In the Attribute Manager, that the Modeling Axis is set to Selected and Orientation is set to Normal.

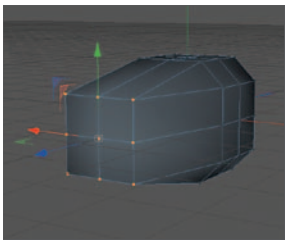
CINEMA 4D offers several options for positioning an object's axes during polygonal modeling. With these settings, the axes will snap to the center of your selected polygons and the Z-axis will face outward, perpendicular to the faces (the "normals"). You now have handy access to the axes for moving points along specific axis without having to return to the toolbar to lock and unlock axes.

Changing the modeling axis does not affect the actual position of the object's axis, as used with the Model mode or for animation.

Select the Move Tool and drag the axis out along the Z-axis.

You have just begun your career as a Polygon Modeler. If you are wondering about the usefulness of such an angular object, remember we have other tools we are not presently using. The *Phong* Tag can make the surfaces smooth and the *HyperNURBS* Object will smooth edges as well as surfaces. For now we will focus on manipulating the polygonal mesh without worrying about the final smoothed appearance.



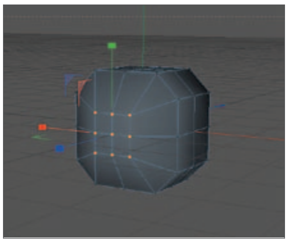


It is also important to get into the habit of using a very important shortcut - a tap of the *Spacebar* will toggle between the most recently used selection tool and the most current function tool such as *Move*, *Rotate*, *Knife*, *Extrude*, etc. Try it now.

Undo the move. Switch to the Point mode and, using the Live Selection Tool, select the points which make up the corners of the four polygons you were just working with (remember the one in the center too). Toggle to the Move Tool (spacebar) and move the points out along the Z-axis.

You should see the same results as before. Let's take a look at two ways to add or remove points and polygons from a selection. First, let's use a selection tool.

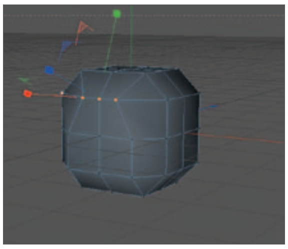
Undo the previous move, the nine points should still be selected. Deselect the center point using the Live Selection Tool with a Ctrl+click on the point.



If you did not use the Spacebar, undo and repeat.

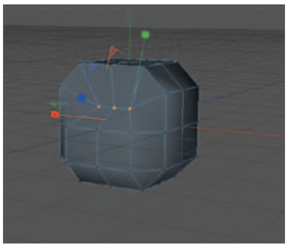
Now add the center point back to the selection using the Shift+click (still with the Live Selection Tool). Return to the Move Tool.

This works the same with polygons as with points. A selection tool is always used for large numbers of points. However, if you want to select or deselect only a few points and are currently using the Move Tool, here is an even quicker method which does not require a switching tools.



Shift+click on the center point to remove it from the selection set. Shift+click on it again to add it.

Although these methods both serve the same purpose, becoming familiar with both will speed up your workflow.



Deselect the center point once again and move the selected points out along the Z (top right figure).

As you can see, this cannot be done with a polygon selection. Polygon selections will always result in movement of all of the corners. You need to work in point mode if you wish to move only some corners of a polygon. Rotate and scale will work the same.

Undo and add the center point to the selection again. Choose the Scale Tool and scale the point selection inward (third from top at right). Undo and select only the top three points of our current selection by dragging the Live Selection Tool over these points without any modifier keys. The new selection of points will replace the old. Scale the selection inward again (fourth from top at right).

You will notice that the reference for the scaling has changed, as evidenced by movement of the modeling axis.

Undo the scale. Change the Modeling Axis from Axis to Object and scale again.

Since the modeling axis is the object's center, the points are oriented inward as well toward the object's center which is now also the modeling axis.

You can also rotate selected groups of points. However, this produces different results, depending on the rotation system selected. The XYZ system works the same as what you experienced above. The HPB system references the object's axis.

Undo the scaling and return to the Polygons Tool.

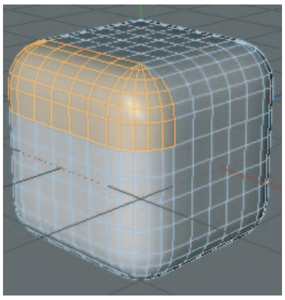
Notice that your polygon selection was recalled independently of any points you selected with the Point mode active. You may also convert a point selection to a polygon selection and vice-versa using the *Convert Selection* command in the Select menu. Let's take a look at some other selection tools.

Deselect all polygons using Shift+Cmd+A.

There are several commands in CINEMA 4D which affect the currently active window so you might get different results, depending on which window is active. *Shift+Cmd+A* will deselect points if the editor window was the last place at which the mouse was active. However, if you last clicked in the Object Manager, the effect will be to deselect all objects.

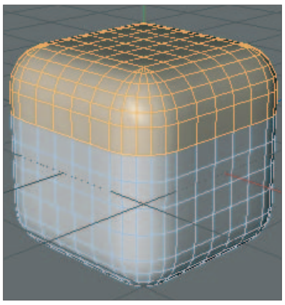
Hide Cube_1, select Cube_2 and switch to the front view. Switch the mode to Gouraud Shading (lines) if necessary. Enlarge and center Cube_2 to take up about 70% of the viewport height by moving the camera.

Make sure you are using keyboard shortcuts as often as possible.



Choose the Rectangle Selection tool. By default, the parameter, Only Select Visible Elements, should be active (check the Attribute Manager). Drag the marquee (selection box) around the polygons that make up the top half of the cube. Switch to the perspective view. Rotate the view to view the cube from all perspectives.

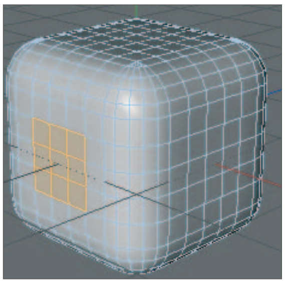
Notice only those polygons which were visible from the front view were selected.



Return to the front view. Disable the Only Select Visible Elements box and again select the polygons in the top half of the object. Switch to the perspective view.

Now notice that all polygons in the top half of your object have been selected. This works the same with points, edges and with the other selection tools.

The Tolerant Selection choice applies to polygons. If it is enabled, any polygon with just a portion included in the selection rectangle will become part of the selection. Otherwise, the entire polygon must be located within the selection area in order to be included.



Deselect all points and make sure you are in perspective view with the Polygon mode selected. Select the Rectangle Selection tool and check the box Only Select Visible Elements. Also, make sure Cube 2 is selected.

We will complete this section with a challenge, allowing you to practice selection and navigation skills. Count and record the number of operations you need to complete this challenge. Before you begin, think for a moment about why you were asked to use the front viewport instead of rotating the perspective view to face the front. You should also know your keyboard shortcuts for changing views.

Select the center nine polygons of all faces, except the bottom.

You should be able to do this using seven operations.

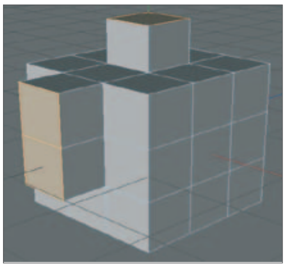
How did you do?

3. Extruding polygons

Complex models require more than just the polygons provided by simply making a primitive editable. In this section we will examine various ways of adding polygons.

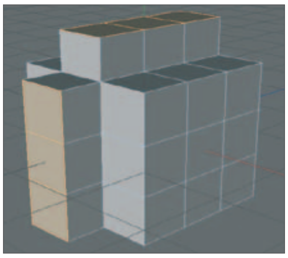
Hide your two cubes. Create a new cube with 3 X 3 X 3 subdivision and make it editable. Name the new cube Cube 3. Make sure you are in the perspective view with the Polygon mode and the Live Selection Tool active (make sure that Only Select Visible Elements is enabled). Change the modeling axis Axis to Object and the Orientation to Axis.

Hopefully, you now know where to find these parameters and functions. We are just getting the axis out of the way to better see the results of some tool operations.



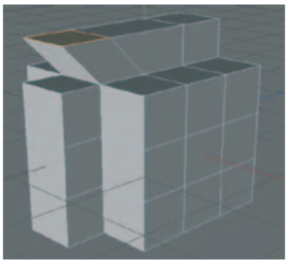
Select the center polygon on the front face and the polygon directly above it. Add the center polygon at the top face to the selection.

Since the first two polygons were adjacent you should have selected them together with a drag. Selecting the isolated polygon at the top requires letting up on the mouse and adding to the selection with a *Shift+click* . If you did not make the selection in this manner, deselect all polygons and practice selecting again.



From the pop-up menu, choose the Extrude Tool.

While you are working on the structure of a polygonal object, the pop-up menu provides quick access to most of the tools you will need. Also pay attention to any shortcut that may listed next to a tool's name in the menu ("*D*" for *Extrude*, for example).

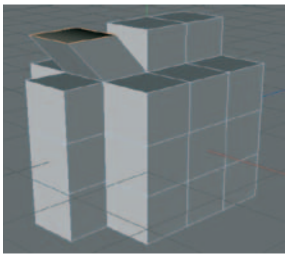


Click and drag anywhere in the editor (if the selected polygons move inward, drag in the opposite direction). When the polygons have been extruded a distance about equal to their width, release the mouse.

New polygons were created to fill the gap left by the extruded polygons. Note also how the selected polygons were each moved along their normals.

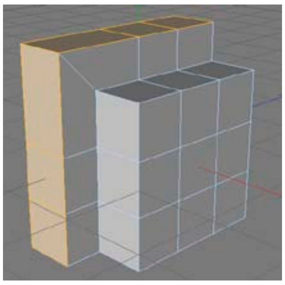
Undo the extrude. Toggle to the Live Selection Tool and select the six polygons running along the center of the front face and across the top. Toggle again and extrude (be sure Preserve Groups is enabled).

You can select with the Extrude Tool active but only by clicking one polygon at a time. Let's see what Preserve Groups does.



Switch to the Move Tool. Select the front-most top polygon and slide it forward along the Z-axis.

The extruded polygons are all still connected, except at the corner, i.e. they still share the same edges. The polygons have been "preserved as a group".



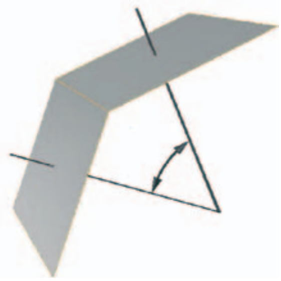
Undo the move and extrusion and uncheck *Preserve Groups*. Extrude the same polygons again.

This looks the same as the last extrusion.

Select the front polygon along the top and move it forward again.

Even though these polygons looked connected, they were not. Each of the extruded polygons has its own four edges and corner points. Generally speaking, you will want to leave *Preserve Groups* enabled.

What if you wanted the extruded polygons along the front face to remain connected to those along the top?

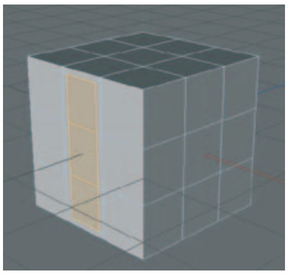


Undo the move and extrusion and re-check *Preserve Groups*, the Maximum Angle value still set to 89°.

You have already done this before, this is just a review.

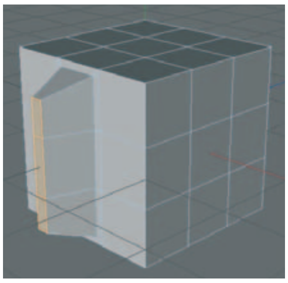
Undo. Change the Maximum Angle for the Extrude Tool to a value of 91° and extrude.

Since the angle between the front and top face is 90° it was not included in the "preserve groups" operation performed earlier. Now that Maximum Angle is set to 91° the front and top polygons are treated as a group and share a common edge.



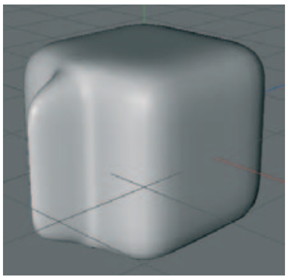
This is a bit different from measuring angles in your geometry class. Here the angle is measured between the normals of the polygons. A normal is the line perpendicular to the plane of the polygon which defines which direction the polygon is facing. In addition to affecting some parameters during modeling, normals are very important in determining how light is reflected from the surface during rendering and how textures are applied.

Sometimes you may want to add more polygons in a certain area without extending the surface outward or inward. The most common method is using *Extrude Inner*.



Undo until your mesh is again a cube and select the three vertical polygons in the center of the front face. Select the *Extrude Inner* Tool and make sure *Preserve Groups* is enabled. Click and drag in the editor window so the selected polygons diminish in size.

Your three selected polygons still share edges but eight new polygons have been created to bridge the resulting gap. You can see that this is a way to build up the polygon count in this area to generate detail.



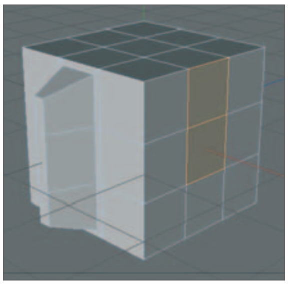
***Extrude inner* once again until your selected polygons are fairly narrow. Using the *Move* Tool, drag outward along the Z-Axis a distance about the width of an original polygon.**

Let's see how this object will look as a HyperNURBS Object. v

Create a HyperNURBS Object and drop your cube into it. Switch to the Model mode so you can better see the object without the polygonal mesh being shown. Rotate the view.

Maybe you can see the beginnings of a fin or similar object...

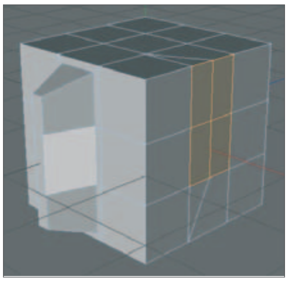
4. Knife tool for subdividing polygons



Now we will subdivide using the knife tool. Try to anticipate the results of using this tool from the discussions in the previous handout.

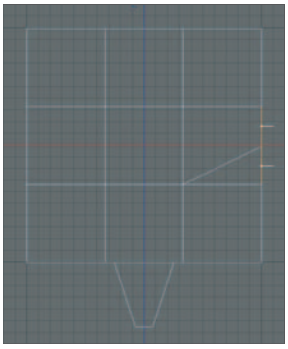
Deactivate the HyperNURBS Object. Make sure the mesh (cube 3) is selected and activate the Polygon mode. Select the center polygon at the right side and the one directly above it as shown at the right. Select the *Knife* tool. Make sure *Restrict to Selection* is enabled in the Attribute Manager. Also in the Attribute Manager, disable the *Create N-gons* box.

Switch to the side view and position the cursor above the center of the top edge of the top-most selected polygon (above the object). Hold down the Shift key and drag the cursor



down past the very bottom of the cube.

Was it really necessary to knife down past the bottom of the entire cube? Predict what the top of the cube will look like.



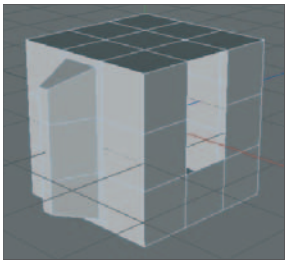
Switch to the perspective view and inspect the top and rest of the cube.

If we leave *Create N-gons* checked, we would not see those triangles. However, CINEMA 4D still would have to replace the five sided polygons with one four-sided and one three-sided polygon (at right). This would have resulted in some asymmetry during rendering. This is because the narrow apex of the triangle pulls the rendered mesh in more than the open areas around it. In many cases it is preferable to have the distortion spread symmetrically.

If you want to subdivide a number of polygons in both vertical and horizontal directions, use the *Subdivide* tool.

Undo the effects of the Knife Tool. Select the Subdivide tool and make sure HyperNURBS Subdivide is unchecked. Leave the Subdivision setting at 1 and click OK

More subdivisions, more triangles. Generally, it is better to create extra polygons using the two Extrude Tools.

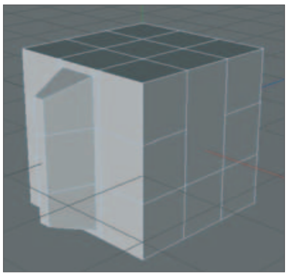


5. Creating and bridging polygons

Here we will give you an idea of how these tools work. We will delete two polygons and replace them using different tools.

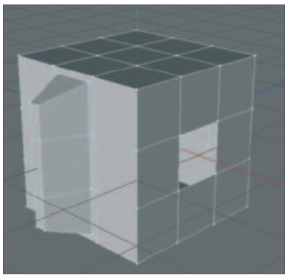
Undo any subdivisions you have just done. Working in the perspective view, select the right center polygon and the one above and delete.

You can now see the interior of the cube through the hole you have created. We will first fill the hole with the *Close Polygon Hole* tool (look under the pop-up menu (*structure*) for these polygon and point tools).



Select the *Close Polygon Hole* tool. Move the cursor over any edge surrounding the hole. When the hole's perimeter is highlighted white, click with the mouse.

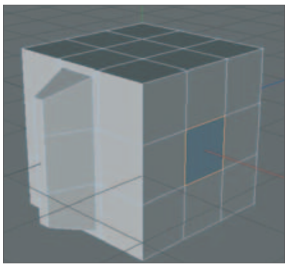
You have just replaced the deleted polygons with a 6-sided N-gon. For many purposes this would be fine. However, as we saw earlier, areas within the N-gon may be broken into triangles that will not give the smoothest results, especially in curved organic objects. The *Create Polygon Tool* offers more control.



Undo the polygon and select the. Move the cursor over the top right corner of the open perimeter. When the point is highlighted white, click with the mouse. Move down to the center perimeter point at the right and click again. Now click on the center perimeter point on the left. Finally, move up to the top left point and double-click.

The double-click lets CINEMA 4D know You are done with that poly. You could have clicked on all six points to make an N-gon as before. You could also have double-clicked on the third point to make a triangle.

The sequence in which you create the polygon makes a difference.



Fill in the center hole using the Create Polygon Tool only, clicking points in a counter-clockwise manner.

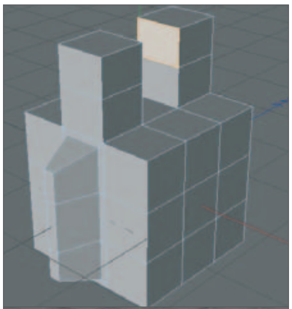
The hole turns a different color when selected. Notice the normal (the tiny white perpendicular line poking out of the center) is pointing away. This could mean trouble when textures are applied.

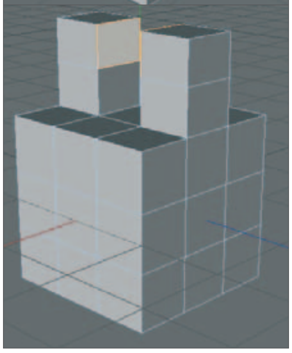
With the reversed polygon selected, choose *Align Normals* from the pop-up.

You do not have to do this with each individual polygon. When you have finished modeling, use *Align*

Normals, with no polygons selected, and all normals will be aligned.

There is yet another tool which can create polygons and more - the Bridge Tool.





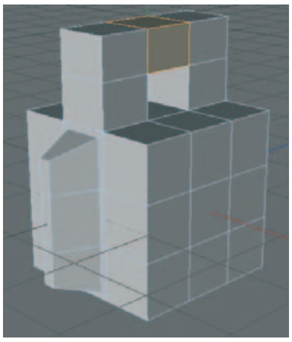
Undo or delete the two newly created polygons. Select the Edge mode (left toolbar) and choose the Bridge Tool. Move the cursor over the upper edge segment along the left side of the hole's perimeter. When the edge is highlighted white, click and drag until the polygon you create ends on the uppermost edge segment to the right (shown at right). Repeat to fill the remaining hole.

The Bridge Tool is quickest at creating polygons in certain situations. Like the two previous polygon creating tools, it can also be used in point mode. However, when used in Polygon Mode, Bridge meets its true calling.

Return to a perspective view and Polygon Mode. Select two polygons on the top face. The one toward the front of the center polygon and the one toward the rear. There should be only the not selected top center polygon between them. Extrude both polygons up about one poly's width. Repeat this so they are standing two polygons high above the top surface. Now select only the two polygons that are directly facing each other across the top center (1st and 2nd views at the right).

Your object should look similar to the figures above. The polygon facing the selected one shown in the figure is also selected.

Select the Bridge Tool and draw a line from a point of one of the selected polygons to a point directly across the gap from it on the other polygon (3rd view on right).



The idea here is that the Bridge Tool is making one edge of a new polygon as you are dragging it. The other edges will be made automatically. The space between the two selected polygons should now be completely enclosed with new polygons and the two previously selected polygons are gone.

Reactivate the HyperNURBS. Carefully inspect the bridge for defects.

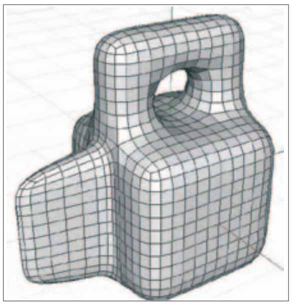
With a little imagination, you can probably visualize uses for this tool.

Besides handles for bottles, the Bridge Tool is invaluable for joining objects such as a hand to an arm.

6. HyperNURBS

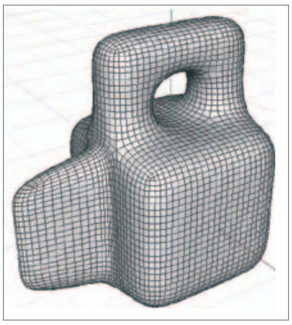
You have already had a taste of what the HyperNURBS function does, let's take a closer look.

Delete (or store) the Phong Tag.



You can temporarily "store" a tag by dragging it to another object. You can drag it back when needed. This saves the time of navigating to it through menus and preserves any parameter settings you have made. Notice the faceted appearance reflecting the amount of subdivision. The amount of subdivision is adjustable in the HyperNURBS dialog.

Select the HyperNURBS and change the Subdivision Editor value to 3.



You now should see much finer subdivisions in your object. You may have noticed, in the Attribute Manager, that separate parameters can be entered for renderer and editor subdivisions. This allows faster redraws in the editor while rendering with higher polygon counts.

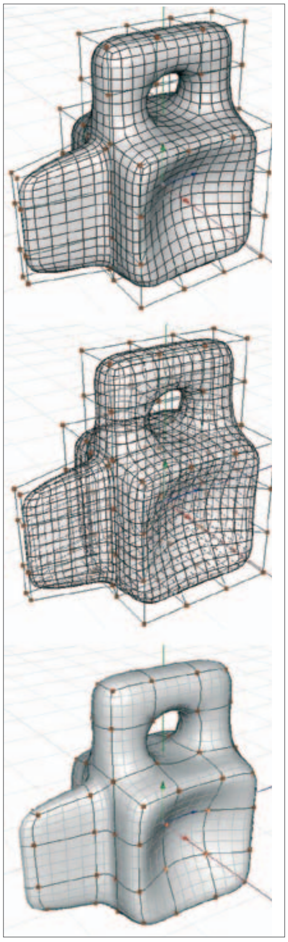
Reset the Editor subdivisions back to 2 and add the Phong Tag back to Cube 3.

Remember, the smoothing (Phong) function only works on surfaces. The edges will still show the facets characteristic of the amount of subdivision you have set.

Look in the Options menu under Options / Level of Detail and make sure Use Render LOD for Editor Rendering is enabled. Move the camera in closer and render in the editor.

This should look relatively smooth with a little noticeable faceting at the edges, good enough for most purposes. Since a higher number of subdivisions is being used, compliments of the HyperNURBS, the edges are smoother as well as the surface. If *Use Render LOD* is unchecked, the scene will render with the lower amount of subdivisions set for the editor (*use Render LOD* does not affect rendering to the Picture Viewer).

From the Options menu bar, choose Level of Detail / Medium.



The object just got chunkier. You see fewer subdivisions just as when you changed subdivision values in the Attribute Manager. The object will still render at the settings in the HyperNURBS. This is useful to speed up editor redraw in large, complex scenes. You can also single out a particular object for higher resolution, while the rest of the scene is displayed with lower detail, by using a *Display Tag*.

Return to the level of detail to High.

Much of the time, especially toward the final stages of modeling, you will probably want to model the cage with the HyperNURBS active.

In the Options Menu again deselect isoline Editing. Select the four points at the corners of the center polygon at the right side of your object. Move them outward along the X-axis.

This mode of operation provides instant feedback of how your object is affected.

Now drag the points inward to create a concavity in the side of your object.

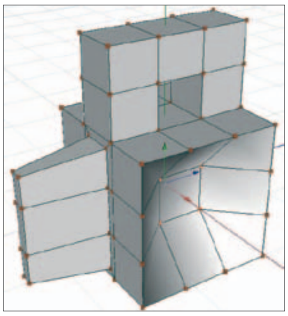
If you have dragged far enough you will notice the selected points have disappeared under the surface. If you went on to edit other points and wanted to select any of these points again it would be difficult to select them. There are three solutions to this problem: First, you can disable the HyperNURBS (left). Another option is to use X-ray mode, available via the Basic Tab in the Attribute Manager (right). X-ray mode, however, looks a bit cluttered because we can see the points on the back side as well (sometimes this is desirable). However, CINEMA 4D offers an awesome third option.

In the Options menu, re-enable Isoline Editing (right).

The mesh is still that of the modified cube you have been working with all along, but it has been projected onto the surface of the smoothed object. You can also see (lighter lines) the subdivisions performed by the HyperNURBS.

Before moving on to deformers we will practice a little point moving. Instead of using undo we will precisely return the points to their original position using the Coordinates Manager

Click on one of the unmoved points on the side face and note its X position in the Coordinates Manager (unless you have rearranged points beyond the scope of this tutorial, the value should be X = 100). Now reselect only the four points you previously moved. Enter the value you noted for the X position of the undisturbed point.

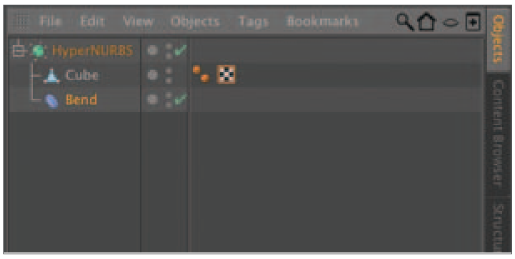


There is not much more to the HyperNURBS Object itself. The details are in the geometry and manipulation of the cage (the polygonal object operated on by the HyperNURBS).

7. Deformers

We will complete our introduction to Polygon Modeling with a brief discussion of deformers. We will visit them again in animation.

Make sure your HyperNURBS Object is activated. Create a Bend Deformation, either from the Command Palette or from the main menu (Create / Deformer / *Bend)*. **Drag** Bend **into HyperNURBS until it is under but at the same level as** Cube 3**.**



The Bend Deformer should be a sub-object of the HyperNURBS Object and on the same level as Cube 3. Since the HyperNURBS only affects the first object in its hierarchy, the cage (cube 3) must appear first (the bend can be a Child of the object to be deformed, more on that later).

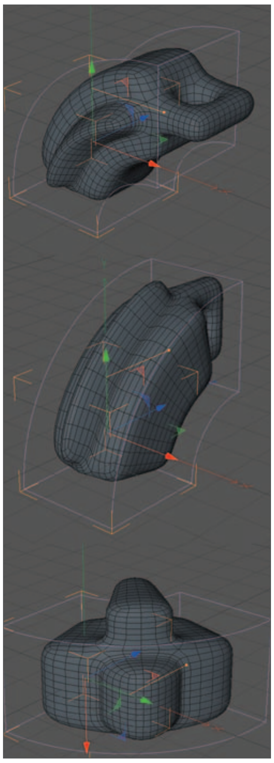
Choose the Model mode. With the Move Tool selected, grab the orange handle and bend the object slightly. Move the mouse around a bit to see how this works.

As with moving anything else in the editor, you can lock axes to control the direction of the bend. More precise control can be achieved in the Attribute Manager.

Undo. In the Attribute Manager, change Bend's *Strength* **to 90 (***angle* **should be 0).**

As you can see from the Deformer's bounding box, it has just bent 90°. You can also set the direction of the bend (angle). The bend always takes place along the Y-axis of the Deformer, so Angle is rather like heading as far as the bend is concerned.

Set Bend's *Angle* **to 90°.**

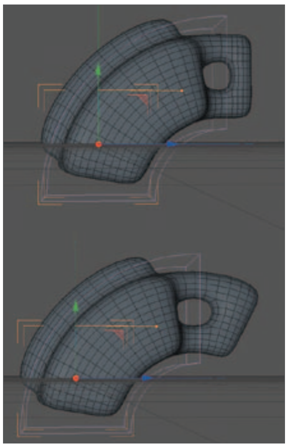


The direction of the bend has swung around 90° in a counterclockwise direction, when viewed down the Y-axis of the Bend Deformer. You could animate the Angle parameter to cause your object to swing in a circle.

If you want the object to be bent in another direction, such as curling up in a circle when viewed from above, you will have to rotate the Deformer.

Select the Model mode and, with Bend **still selected, change the bank angle (b) of** Bend **to 90° in the Coordinates Manager.**

Many deformers, including bend, have a *Mode* setting that specifies how any part of the object outside the Deformers bounding box will be treated.



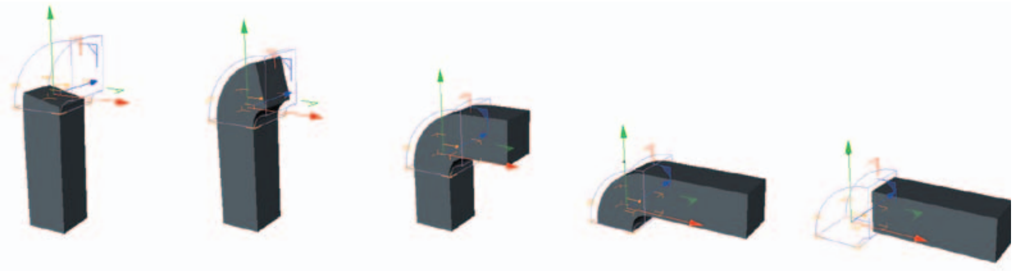
Reset Bend's **bank to 0. Rotate to more of a side view and study the object for comparison.**

Notice that the top portion of the object which sticks outside the Deformer's bounding box is not bent. This is the behavior of the Limited mode.

Change the Mode setting to Unlimited and observe the difference.

The bend keeps on affecting the object beyond the limits of the bounding box. The Limited mode is useful when bending a character from the waist only. Used with the *limited* mode, Bend can also be animated to move down an object, causing a wave-like motion.

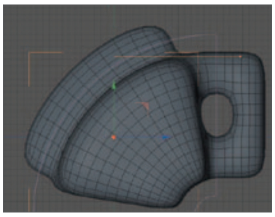
Change the Mode of Bend back to Limited. With the Move Tool selected, grab the Y-axis of the Deformer and drag it up and down.



You should see a similar transition happen to your object, as pictured above.

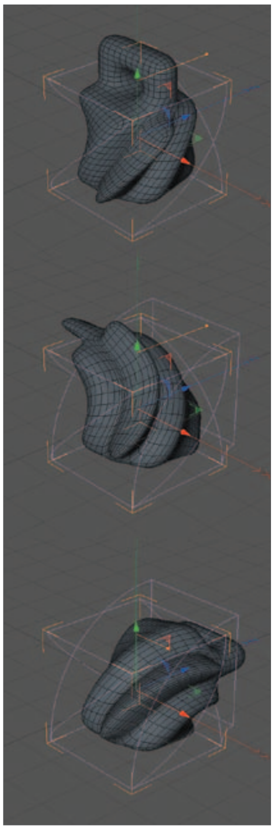
You may have noticed that the object seems stretched by the bend. This is not always desirable. You probably would not want a snake getting longer just because it was slithering around a corner.

Check the **Keep Y-axis Length** box.



Many different, seemingly unrelated effects, can be achieved by re sizing, rotating and moving deformers. Deformers are used in animation but they can also be used temporarily to help create a polygonal mesh to be used in a Hyper-NURBS Object. In the main menu Mesh / Conversion / Current State to Object, if used on our bent object, would create a mesh just like that in the editor, only without using any deformers. Although we have been practicing on a polygonal object (within a HyperNURBS), Deformers work equally well with primitives as long as the primitive is given enough segmentation.

Now let's take a look at another deformer, the Twist Deformer.



Deactivate and hide the Bend Deformer. Create a Twist Deformer and place it under Bend (at the same level). Drag its interactive handle until you have a twist of about 90°.

You can judge this by eye or by observing the *Angle* parameter in the Attribute Manager as it changes dynamically.

Set Twist's height (y Size) to 200 and Angle to 90°.

Note again that the upper portion of the object sticking out of the Deformer's bounding box is not twisted. This is due to the default mode of *Limited* . Let's see what two deformers do together.

Change the mode of both deformers to Unlimited. Reactivate Bend.

There is no limit to the number of deformers you can use with a particular object. You may, for example, wish to use several bends affecting different regions of a single object. Remember, the order of the Deformers makes a difference.

In the Object Manager, drag Twist **until it is just above** Bend**. Alternate the order of** Bend **and** Twist **using undo and redo (***cmd-Y***) and compare the results from various viewing angles in the editor.**

Before moving on to other types of deformers, let's review two other properties of deformers. Do not forget that deformers can be used on any type of object, including Splines. Also, the Deformer can only work at the resolution of the underlying object. If an object has only two subdivisions along the Deformer's Y-axis and a Bend Deformer is applied, the object will appear folded at the center. If there are 20 subdivisions, the bend will consist of 20 segments and will appear much more rounded (review this from the last tutorial).

Three other deformers work a bit differently.

Disable Bend and Twist and create an Explosion Deformer. Make it a Child of HyperNURBS (under the mesh again) and drag its interactive handle (at the origin by default).

You can also define the explosion's *Strength* value in the Attribute Manager. This Deformer is almost always used for animating. *Melt* and *Shatter* work much the same way. The *Explosion FX* Deformer is quite a bit more complicated but can make the fragments look solid.

Deformers can also be placed as Children of the mesh they affect instead of at the same level within the HyperNURBS. In most cases, such as with bend, it makes only a slight difference and can help tidy up the Object Manager. However, when using Explosion, and perhaps others, it does make a difference.

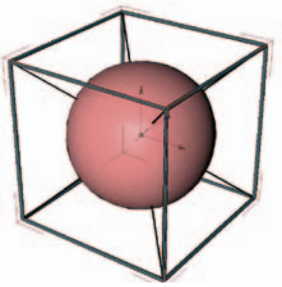
Make the explosion a Child of the cage (cube 3) and drag the interactive handle.

Notice the difference in the exploded segments. As a Child of the polygonal object, the object is exploded into as many segments as there are polygons. However, when it operates on the HyperNURBS instead, it uses the HyperNURBS subdivisions to create the fragments.

Most of the other deformers work in much the same manner. The exceptions are the *FFD (free Form Deformation)* and Joint deformers. These have special applications to animation, they are used to make objects change shape over time such as a human or cartoon character walking or talking (character animation). Hope you had fun!

Polygon Practice

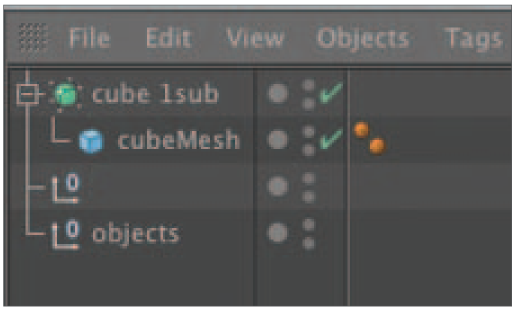
Polygonal modeling is the most tedious and difficult of the modeling techniques. However, it is an essential technique for producing most of the sophisticated models appearing in a scene. Almost any organic model such as plants, animals, humans and cartoon characters are created using polygonal objects combined with HyperNURBS. In the following examples we will practice different methods of creating and modifying polygonal meshes. Except when new tools or techniques are introduced, the details will be kept to a minimum. You may need to review previous tutorials if you have forgotten a particular technique.



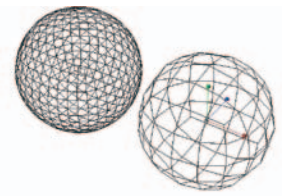
The cage (polygonal mesh)

Before modeling some shapes, it is important to get an idea of the relationship between a polygonal mesh (the cage) and the HyperNURBS Object used to smooth it. As you proceed through this intro, think of the cage as a scaffolding. The HyperNURBS Object suspends the surface from that scaffolding using elastic bands. However, sometimes the bands seem tighter. Let's see if we can deduce what factor affects the tightness from some examples.

Open a new CINEMA 4D file, make any standard adjustments, set the display mode to Gouraud Shading and save the file in the "Intro Tutorials" folder as "Polygon Practice". Create an objects object in which to store your objects as you complete them (and a "spacer Null Object" if you wish). Create a Cube Primitive named cubeMesh and drop it into a HyperNURBS. Name the HyperNURBS Cube_1sub.



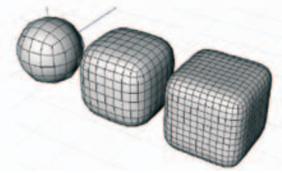
Notice the HyperNURBS sphere appears a bit chunkier than the Sphere Primitives you have previously created (create a sphere now for comparison if you need a reminder). Remember, Sphere Primitives were made finer by adjusting the segmentation. You can do this with HyperNURBS Objects as well.



In the Attribute Manager, change Cube_1sub's Subdivision Editor value to 4.

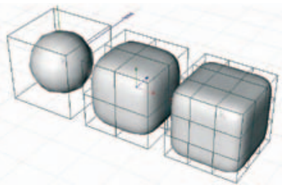
This should look quite a bit smoother. Since this is just the editor, we will stick with low subdivision to speed things up.

Change the display to Gouraud Shading (lines) (with Wireframe) and note the subdivision. Change the editor subdivisions back to the default of 2 and again note the subdivisions. Make cubeMesh editable.

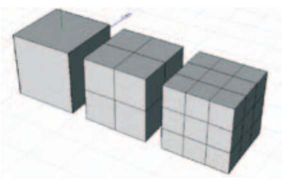


Making the cube editable should have made no difference in the appearance of the object. However, now that the cube is composed of points and polygons, you can modify it in many ways to produce an infinite variety of shapes, even having started with this simple six polygon cube.

Create a new Cube Primitive and drop it into another HyperNURBS. Name the HyperNURBS Cube_2sub and move it to X = 250. Set the Cube's subdivisions (segments) to 2 for each axis and make it editable. Create a third cube with three subdivisions for each axis and make it editable. Drop it into a third HyperNURBS (with an appropriate name!) and move it to X = 500. Rotate the view a bit toward the front to see all three objects completely (shown at the right).



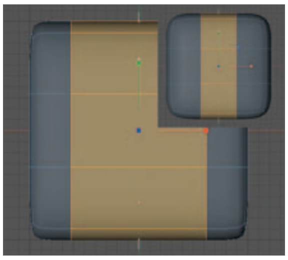
The HyperNURBS subdivision is the same for each object. Why do the cubes appear to have different subdivisions? Because the underlying mesh is subdivided differently (right). The HyperNURBS takes each subdivision and subdivides it once and then once more (the value 2 in *Subdivision Editor*).



Choose the Polygon mode and click on the mesh for each object. Study the relationship between the cage and the shape of the surface (right).

Before going on, make a prediction about the appearance of a similar object with 10 subdivisions for each axis. Most importantly, make a statement about the size of the polygons (or the spacing of the points) and the closeness of the surface to the cage.

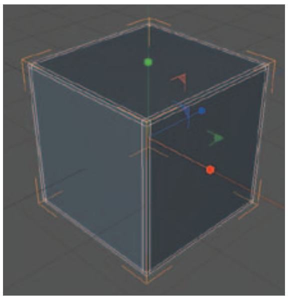
Create a Null Object, name it Cubes and drag your three objects into it. Drag a copy of Cube 3sub out of Cubes and name the new copy The Cube. Drag Cubes into objects and hide objects. Move The Cube back to the World origin.



Have you thought about how we will turn this object into a cube with no additional points. Think about it before proceeding.

In the front view, select the Gouraud Shading mode. Select the center vertical row of polygons (back, top and bottom too). Using the Scale Tool, stretch the polygons outward along only the X-axis until the edges are almost over the outside edges of the mesh.

Those edges sharpened up quite a bit. If you scaled right out to the edge, the edges would get quite sharp. You could also achieve the same results scaling the points outward instead of the polygons.

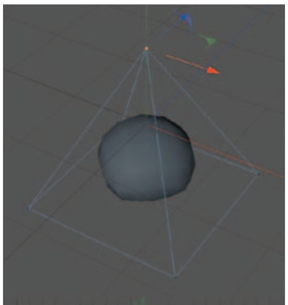


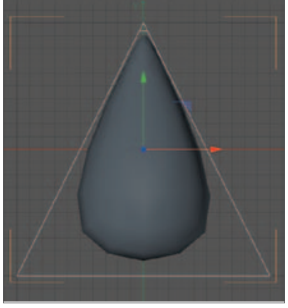
Do the same operation vertically on this face and once again on another face such that all edges have the same sharpness.

Of course you can build this more easily with a primitive but the point here is to emphasize the relationship between point spacing and how the surface is suspended from the mesh. There are plenty of occasions you will want an otherwise curvy object to have some sharper edges.

Primitive polygon modeling

You have just seen how to model a sphere and cube using polygonal meshes in HyperNURBS Objects. Let's model some more primitives to gain practice in technique. Again, pay attention to how the cage is manipulated to produce certain shapes and features. As you read the objective for each object to be created, try to think of the process before reading the instructions.





Store The Cube **within objects. As you finish each of the following, name it the same as the bold heading (Cone, Cone 2, etc). and store in objects.**

Cone

Let's model a HyperNURBS cone from a Cube Primitive.

Create a default Cube Primitive, make it editable and drop into a HyperNURBS. Choose the Point mode and select the cage.

A cone has only one point at its apex, our cube has four. Let's combine those four into one point with the *Weld* Tool.

Select the top four points and choose Weld from the pop-up menu. Click on the yellow center point.

A click on one of the corner points would have placed the welded point there. Let's sharpen up the top.

Think back on how to sharpen edges.

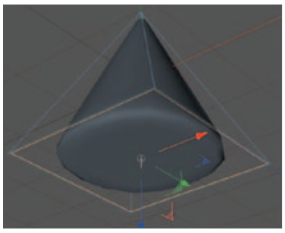
Switch to the front view and select the Knife Tool (pop-up menu or K). In the Attribute Manager, uncheck Visible Only. Use the Knife Tool with the Shift key pressed to create a horizontal row of points just under the top point.

The *Shift* key constrains the motion of many tools horizontally or vertically (and more if you wish). The closer to the top point you used the Knife Tool, the sharper your apex will be.

You can move these points down or up later to soften or sharpen the point. Remember, most real-world objects lack the sharp edges and points which computer images often produce.

Back in the perspective view, in Polygon Mode, select the bottom poly. Perform a very slight Inner Extrude (to do this easily, set the Offset to 1 and click Apply after selecting the tool).

You also could have horizontally knifed very near the bottom from the front or side. The bottom's still a bit rounded...



Do something to sharpen the edge. Inner extrude again, or try the knife from the front.

More rows of points or polygons clustered together will produce sharper edges.

Cone 2

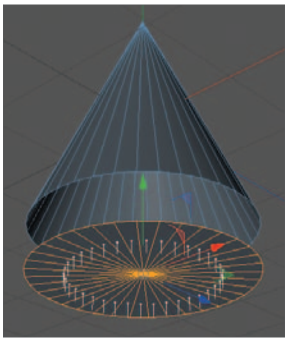
Here we will model the HyperNURBS cone from a cylinder primitive.

Create a cylinder primitive, change the Radius to 100 and the *Height Segments* to 1. *Make Editable* and drop into a HyperNURBS.

This didn't seem to get all rounded by the HyperNURBS like the Cube Primitive did.

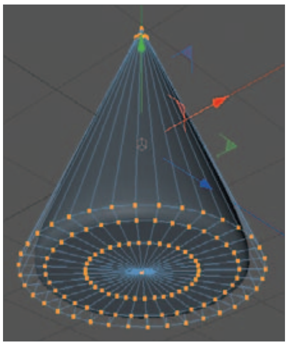
Weld all the top points (when selecting the top points, make sure *Only Select Visible Elements* is unchecked). Sharpen the top as before.

Well, this certainly seems easier than building it from a cube. However, these objects are different in a couple of important ways. First, this one is made of quite a few more polygons than the cone above. There is another, more important difference.



In the perspective view, select only the bottom polygons. Move them down along the Y-axis.

Did you wonder why the cone made from the cylinder already had a flat bottom? Unlike the cube and sphere, most other primitives, such as the cylinder, use caps for their ends. When made editable, these caps are not connected to the sides. The points of the outside circle making up the bottom sit exactly on top of an identical set of points making up the bottom row of the sides. This makes the edge as sharp as can be. So this is why the cylinder didn't get created to 360° when dropped into the HyperNURBS Object.



Although having the cage consist of unconnected polygons to yield sharp edges sounds good, it is not usually desirable, especially for texturing and rendering. Except in special circumstances your mesh should be continuous. This generally means having to begin with a cube or a sphere.

However, here is a way to make your mesh whole.

Return the bottom polygons to their original position with Undo. Switch to point mode and make sure all points are deselected. From the pop-up, choose Optimize. Check the Points option before clicking OK.

Your cone should now have a rounded bottom. All points within 0.01 units of one another (the default Tolerance value for Optimize) have been welded together. Now the sides and bottom share points in common so the cage is now made of polygons with no disconnected segments.

At this point in your modeling experience, you should be able to state two techniques to make the bottom edge relatively sharp again while maintaining the continuous mesh you just produced by optimizing.

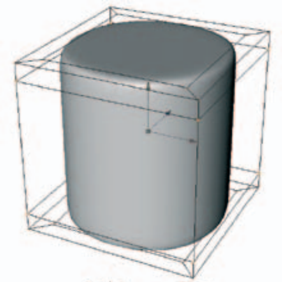
In this case, you could also select the two inner circles of polygons on the bottom and scale them outward until the outer circle became very thin.

Undo the optimize before storing.

Cylinder

Next we will use a Cube Primitive to model a HyperNURBS Cylinder. You might be able to figure this out on your own after the practice you have had already.

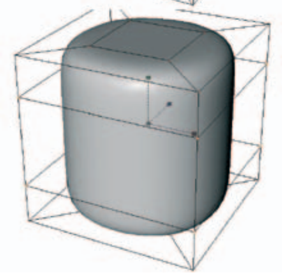
Drop a Cube Primitive into a HyperNURBS and make it editable. Select the top and bottom polygons and perform two slight Extrude Inner functions.



Rounded cylinder

Here is a slight variation on the technique above which allows more control in rounding the edges.

Drop a Cube Primitive into a HyperNURBS and make it editable. Select the top and bottom polygons and Extrude Inner by 10 units. From the front view, knife a horizontal row of points about 20 units below the top row and another 20 units above the bottom.



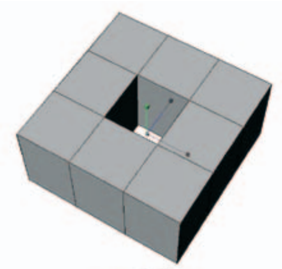
You can adjust the rounding by carefully selecting and scaling the points.

From the front view, select the two new horizontal rows of points and scale them inward along the Y-axis only. From the top view, select all the inside corner points from the inner extrusion and scale them inward with the X and Z-axes active but the Y-axis locked.

Disc

It is pretty easy to make a HyperNURBS disc from a plane primitive.

Make a HyperNURBS disc from a plane primitive.

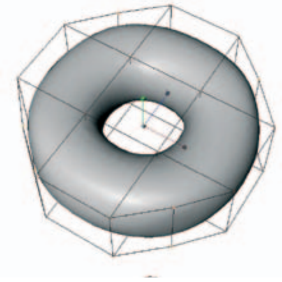


Does your disc look more like a slightly rounded rectangle? Make the plane editable (or view in wireframe mode) and look at the polygons to determine why.

Torus

To finish this section we will make a torus the hard way in order to practice a few techniques.

Create a Cube Primitive with a Size of 200, 80, 200 and Segments 3, 1, 3. Make it editable.



Before placing into a HyperNURBS, we will create the hole.

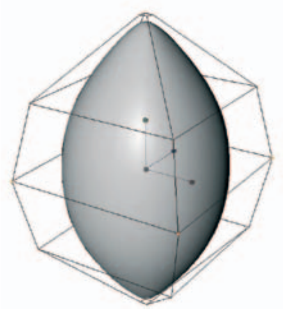
Select the top and bottom center polygons and switch to the wireframe view. Using the Bridge Tool, drag a line from one corner of the top selected polygon to the corner of the bottom polygon, which is directly under it in the Y direction (right). Now drag the mesh into a HyperNURBS.

Looks like the corners need a little rounding.

Switch to the top view and select the corner points (make sure to also select those on the bottom face). Using the Scale Tool with the Y-axis locked, bring the eight points toward the center until you have achieved a more circular appearance. Inspect from the perspective view.

It probably looks a bit puffed near the points which were just scaled inwards. This is because they are crowded a bit closer to the center points than the center points are to each other. You could further adjust these to produce a more perfect torus if necessary.

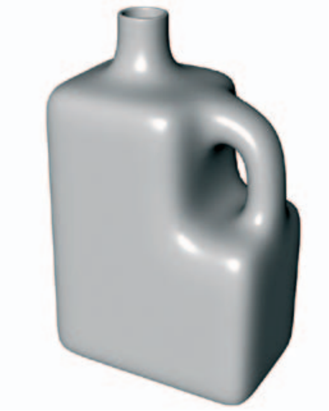
Football



Model the football based on the mesh shown at the right.

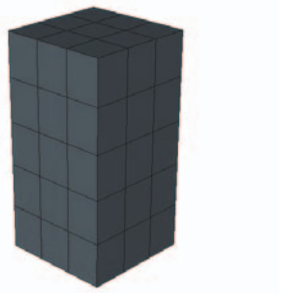
Although not too inspiring creatively, hopefully this tutorial has given you more insight into how the geometry of the mesh affects the final appearance of a HyperNURBS Object. We will begin to create some real world objects next.

HyperNURBS Bottle Tutorial



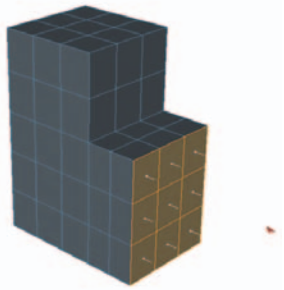
In this tutorial you will create a polygonal mesh and place it in a HyperNURBS Object to produce a plastic bottle with a built-in handle. More importantly, you will practice techniques in mesh building and see how the placement of points affects the appearance of edges.

Create a new folder in your student folder titled "Plastic Bottle". Open and set up CINEMA 4D. Save the file as "Plastic Bottle 1" in the new folder. Create a cube and name it botMesh. Size the cube to 200, 400, 200 and set the segments to 3, 5, 3. Make the cube editable and select the Polygon mode.



Use of a cube is probably the most common way to begin building a cage for a HyperNURBS Object. We will add more polygons to this basic cube as needed. We are also going to practice a couple of strategies to name and backup work as we proceed. The name botMesh tells us this is the polygon mesh (cage) for the bottle. We will reserve the name Bottle for the HyperNURBS Object.

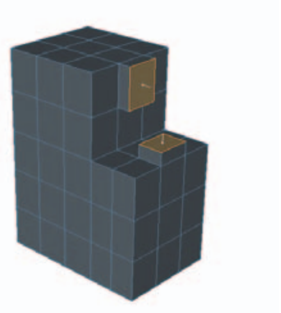
Select the bottom nine polygons at the right side (facing along the X-axis) and extrude them twice, about 50 units each time (as shown at the right).



Remember, if you need precision on the extrusions, you can set a specific value in the Attribute Manager in the *Offset* text box. As is generally true, you also want the *Preserve Groups* option checked to keep your nine polygons joined together.

Select the two polygons and extrude them as shown.

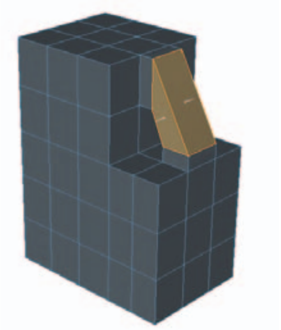
You will notice that each of the two polygons are extruded independently of one another in the direction of their normals (the only reason to extrude them one at a time is if you wanted to extend one more than the other. However, even after extruding both you could select one of the polygons and perform a *Normal Move* on it individually).



In this next step you are going to build a handle by filling in the space between the two selected polygons using the *Bridge* Tool.

With the two polygons you have just extruded still selected, choose the *Bridge* Tool. Drag a line from the corner of one selected polygon to the corresponding corner of the other polygon (such that the line defines an edge of one of four new polygons that will bridge the gap).

The two original polygons have been deleted and the space between them filled by four new polygons. We are going to take a break from developing our model here to perform two types of back-up.



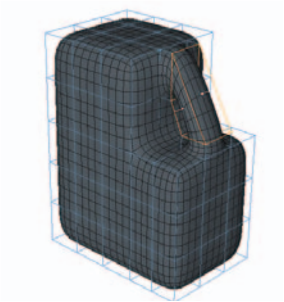
Save the file (*cmd+S*) as you should be in the habit of doing frequently.

On larger projects, you will want to save multiple copies of the scene as you develop it ("Plastic Bottle 2", "Plastic Bottle 3", etc). This lets you return to an earlier state when, as often happens, some line of development does not work out or something catastrophic happens to your current scene. This can be achieved by using the *Save Incremental* item under the file menu.

Next, create a Null Object and name it "...bk..". Drag ...bk.. so that it is at the bottom of the object list and hide it (editor and renderer). Drag a copy (Control+drag) of botMesh down into ..bk.. Leave the copy alone in the back-up Null Object to return to if necessary.

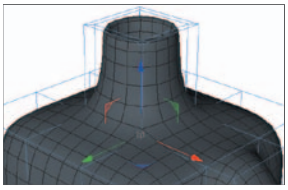
Now you can continue work with the copy outside the back-up Null Object and you have a back-up in the Null Object to return to if necessary. As you proceed with modeling, drag a new copy into the ..bk. from time to time, especially before experimenting. As you drag successive copies into the Null Object, the most recent will be stored at the top of the list. You will find this more versatile and reliable than depending on undos to return to a certain state of modeling. However, all of this is wasted if something crashes loosing all your changes since the last save. Solution: save often (in other words, no crying about what you lost when the power went out!)

Now you have not only backed up your file to the hard drive with a save, you have saved the current state of your object in the "..bk.." Null Object so you can rapidly return to it if you do not like the progress on the active version. Back to our bottle.



Make your mesh the Child of a new HyperNURBS Object. Rename the HyperNURBS Object Bottle.

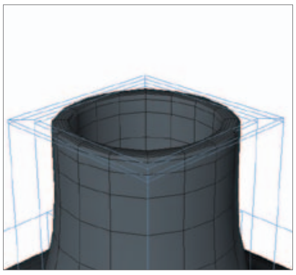
Your object should now look a lot like the smooth form pictured to the right. We will work a bit more on the handle and overall proportions later, let's build the spout next. This will take three extrudes: 1) up from the top surface; 2) an inner extrude to give the spout thickness; 3) an extrude down into the bottle for the inside of the spout.



Select the top center polygon (ignoring the handle) and extrude it upward about 80. Next, with the extruded face still selected perform an Extrude Inner of about 6. Finish with an Extrude of about -100 (in toward the inside of the bottle) to give the appearance of the spout leading into the inside of the bottle.

Your spout should look as shown. You will notice that the rim around the spout appears too thin and sharp. Let's thicken it up a bit.

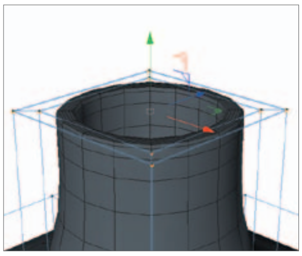
Drag another copy of the cage into the ..bk. Null Object. Switch to the side view and move the camera in. Make sure the Polygon mode is active and choose the Knife tool. Make sure Visible Only is unchecked. Deselect all polygons using Cmd+Shift+A (the deselect command also works in the Object Manager so be sure the editor window is currently active). Create new polygons just barely under, and parallel to, the very top edge of the spout. Use the Shift key to keep the knife horizontal (two lower figures to right).



You have now created eight new polygons, four on the outside and four on the inside of the spout. These are thin polygons with their points very close together vertically.

Use the perspective view to see the results.

You should now notice a distinct flatness to the edge at the top of the spout suggesting some thickness to the plastic. Edges can be sharpened by placing points close together. The closer together the points are, the sharper the edge. Let's play around with the edge sharpness to better see this.



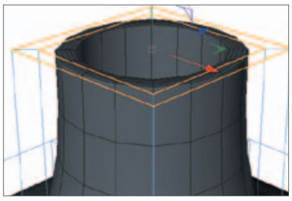
Select only the eight new points you created by knifing. Do this in point mode from the side or front view using the marquee selection tool (be sure *Only Select Visible Elements* is still unchecked). Switch back to the perspective view, select the *Move* Tool and move the selected points up and down in the Y direction and observe the results.

You used the side view to select the points because it lines up the points on the backside with those on the front so that you can be sure they will also be selected. In the figure at the right, the row of points has been moved up until it is almost on top of the top row.

When the points are moved down to increase the separation between them and the very top points, the top edges become more rounded. When the points are moved up until they occupy the same position as the top points, the edge becomes very sharp. Leave your points where you think the spout edge has the most natural rounding.

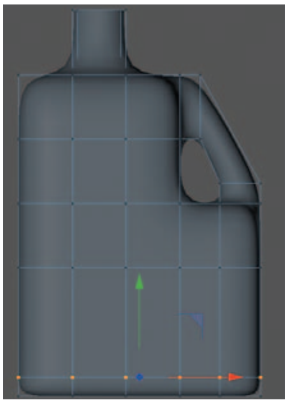
Here is another way to sharpen the spout using HyperNURBS weighting.

Rename the cage botMesh.knifed. While in ..bk.., Control Drag a copy of the most previously saved (topmost within ..bk). botMesh out of ..bk. and into the HyperNURBS. Name this copy botMesh.weighted. Drag botMesh.knifed into ..bk..



We will use HyperNURBS weighting to cause the edges around the top of the spout to pull more strongly on the mesh.

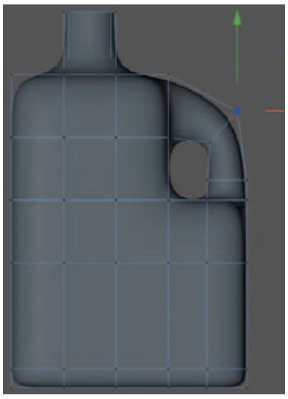
Choose the Edge mode and select the four outside and four inside edges at the top of the spout (but not the tiny edges dividing the corners of the top polygons). Hold down the period key and click and drag to the right. Adjust by dragging either way until the edge has the desired sharpness.



When the *Live Selection Tool* is active you can also adjust the weighting in the Attribute Manager under the HyperNURBS tab. Weighting can also be done on points and polygons (see Help files). Weighting can reduce point and polygon counts but does not always give the best results. Try it as the need arises and resort to adding polygons, if necessary.

A typical bottle like this might not be as rounded on the bottom as ours so let's take a little of the curvature out down there.

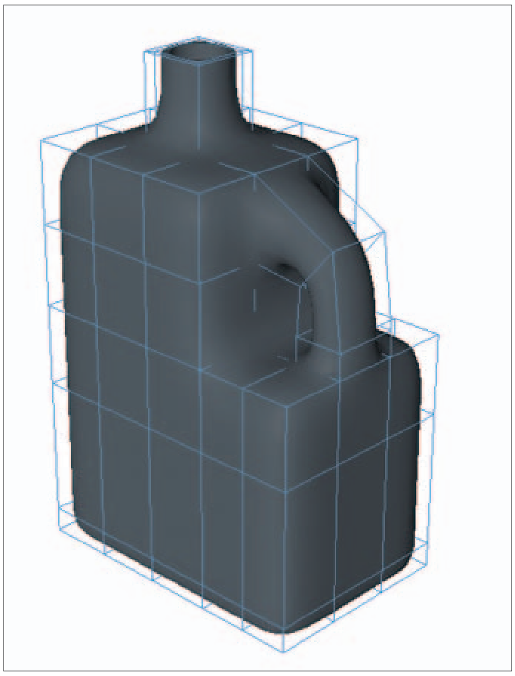
From the side view, select the second row of points from the bottom (being sure to select those on the backside as well). Move them down in the Y direction until you are satisfied with the shape at the bottom.



The handle is looking a bit awkward - like it is pinched in the middle. We are at the stage where we will have to move around a few points a pair at a time to adjust the look.

Also from the side view, zoom the view for a closer look at the handle. Choose the Live Selection Tool and uncheck Only Select Visible Elements. Select and move the points around until your handle appears a little more uniform (right).

Unchecking the *Only Select Visible Elements* box enables you to select both the visible point and the one behind it on the other side of the handle. This way you maintain symmetry of the handle while moving points. Also, moving from the side view keeps your points in the ZY plane as you move them even if you forget to lock the X-axis.



Save another copy of your mesh in ..bk.. Save the file again as well.

We will revisit this bottle later to texture it and add a label. You could now further tweak your bottle's shape if you wish. Maybe the spout is too long for your taste. The spout might appear to come out of the top with too much roundness. Try some adjustments if you have time and save any further changes to a new file. Think about some ways to put threads on the spout, this is what 3D modelers get paid the big money to do.

Table Modeling Tutorial

In this tutorial you will model a wooden table. You will use different techniques to achieve similar results and then compare these techniques.

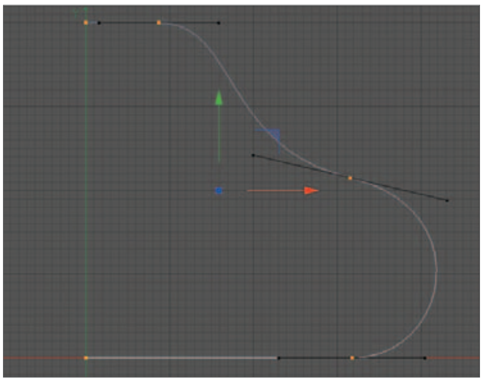
We will begin by modeling the tabletop. Different table designs may best be achieved by different techniques. For example, a round table made entirely out of wood would probably best be done as a Lathe NURBS Object. A table made of two materials, such as a Formica table edged in aluminum out of the 1950's, might better be created as two objects. We will build a rectangular wooden tabletop with rounded corners and a classic edge profile. First we will build the top with sweep and Loft NURBS Objects, then we will build the same top using a polygonal mesh inside of a HyperNURBS Object.

The sweep and loft top

Create a new folder, "Wood Table". Open and set up CINEMA 4D. Save the file as "Wood Table 1". Create a Rectangle Spline Primitive. Set its Width to 300 and its Height to 600. Check Rounding (leave the default Radius value at 50) and change the Plane to XZ. Name this rectangle topPath.

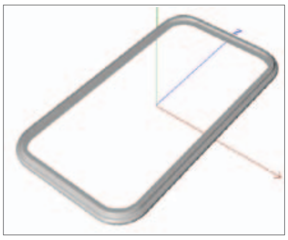
Had you created your Spline in the top view, it would have naturally ended up in the XZ plane. However, in the perspective view, the default is the XY plane, as if you intend to view it from the front. The Spline topPath will become the path for a sweep object that will define the edge of our table. Now let's create a profile to sweep along the path.

In the front view, draw a Bézier Spline as closely as you can to the one pictured at the right. This Spline has five points. The two on the bottom must have their Y values equal to zero. The top two should have Y values of exactly 20. Both the top and bottom points on the left must have their X values set to zero. Also, tangents of all points except the one in the middle should be absolutely horizontal. Do not close the Spline. Name the Spline topProfile.



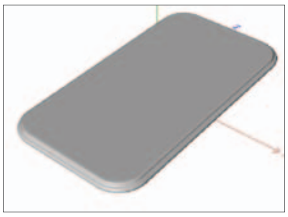
This attention to detail with the point positions and tangents is necessary because we will need to fit the table's edge, of which this profile is a part, to the top and bottom surfaces, which will be separate objects. Attention to detail at this point will make the rest of the task easier.

Create a *SweepNURBS* Object and name it edge. Drag the two Splines into edge; make sure the profile is above the path in the hierarchy (and at the same level).



You have now defined the edge of your tabletop. We need a top and bottom surface that fit perfectly into this border.

Create a *Loft NURBS* and name it bottom. Drag a copy of topPath into it.



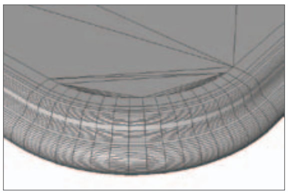
When only one Spline is used with a loft object, the Spline is turned into a flat surface. You have just created the bottom surface of your tabletop, now you will create the top surface.

Copy bottom and rename it top. Change top's Y position to 20 to coincide with the top points of the edge.

Since the same Spline was used for the edge's path and the two surfaces, everything fits perfectly. It is time to organize these three components into a group. You could create a Null Object and drag each object into it. However, there is a faster way:

Select all three objects and hit Option+G. Name the new parent Null Object TableTop_high.

Making a lower resolution bersion



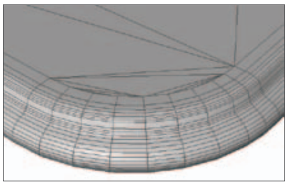
While in the perspective view, switch to Gouraud Shading (lines). Enlarge the view to show the corner in detail as pictured to the right.

The lines in "Lines" with "Wireframe" display modes show you how a Generator object will be converted to polygons when rendering. Those long, skinny triangles on the top would present problems in some texturing situations like a mirror's surface. Here they will pose no problem.

You will notice the edge, as it goes around the corner, is divided into 18 segments. This is due to the default interpolation (*Intermediate Points*) used to make the path. The *Angle* value was set to 5° which means it places a point every time is goes around another 5° (resulting in another polygon). This makes sense: 18 turns X 5° per turn = 90° around the corner.

Render and note the smoothness of the corner.

The default smoothness looks pretty good, even this close. If this were to be a large animation with lots of objects, this smoothness would slow us down. Let's assume our table will not be this enlarged nor the center of attention. Let's lower its polygon count for practice.



Make a new copy of TableTop_high and name it TableTop_low. Hide TableTop_high and do the following to TableTop_low. Change the Angle values for the interpolation of all three Splines from 5° to 10°.

Hopefully, you saved time by selecting all three topPath Splines so you could edit them together.

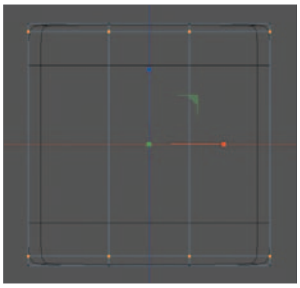
Render and note the smoothness.

This should appear definitely less smooth. However, once textured and placed in a complex scene, will render substantially faster.

Back the view away until you can just see the entire top and render.

The difference in resolution is not noticeable from this far away. Although not really a necessity for this small project, you should remember and practice this style of efficiency in large scenes or animations that consist of more than just a few frames.

A HyperNURBS top



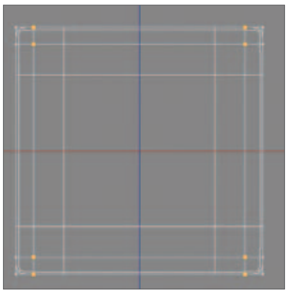
Now we will make a similar top using a HyperNURBS Object.

Hide TableTop_low and create a new Cube Primitive named topMesh. Set the size to 350, 20, 350 and the number of segments to 3, 5, 3. Make the object editable and place it in a new HyperNURBS Object named TableTop_2. Select the Point mode and switch to the top view with Gouraud Shading.

We are creating a square tabletop for reasons to be explained later. Notice how the corners are more elliptical than circular in their rounding.

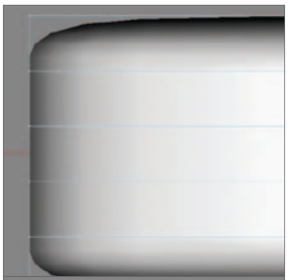
In the top view, select the two center horizontal rows of points (including the back). Using the Scale Tool, move these points away from each other along only the Z-axis (right). Now grab the two center vertical rows and scale outward along the X until the corner polygons are square (bottom right).

Notice that the rounding looks nice and symmetrical when the points create square polygons at the corners. If you want a sharper corner you will need to scale both the horizontal and vertical rows further toward the ends, keeping the corner polygons square.



Now we will work on the edge. The selection option *Only Select Visible Elements* should remain unchecked throughout all of this (hopefully you did that in the procedure above to also select the points on the back). Zoom in to the edge while working on the tabletop. To avoid switching back and forth between views, we will add a new view panel to our screen.

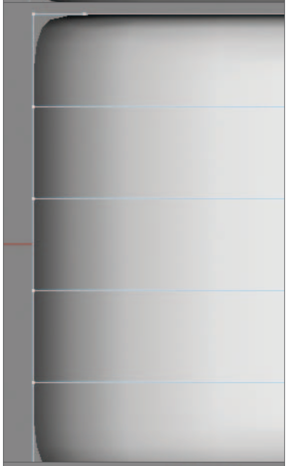
From the main menu choose *Window / New View Panel*. Dock the new viewport by grabbing its shaded rectangle icon and dragging until the cursor is over the right half of the "Object" tab. In this new viewport, change to a front view with Gouraud Shading and move the camera in until you have a nice fat view of the edge in profile (shown at the right).



You can click on the "Objects" tab to return to the Object Manager at any time. The active window is highlighted by a thin blue line. Select points from the main window, see the detail in the new window. Any changes you make will appear in both windows at once.

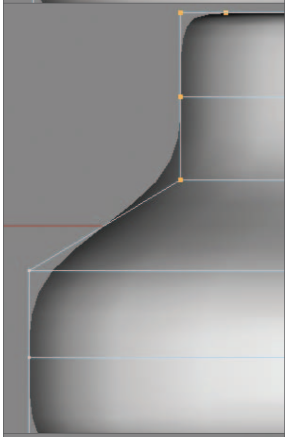
The way the top gradually slopes down toward the sides (top right) is not very realistic - we should fix that before working on the edge.

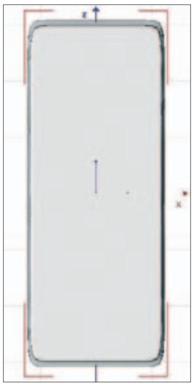
Back in the main view panel, while still in the top view, select all top and bottom polygons. *Extrude Inner* just enough to flatten the top while allowing for a little curvature at the corner (figure to right). Use the new viewport for feedback as you drag.



Much better. Leave the new view panel as is to view your edge up close while you perform selections and movements in the main view panel.

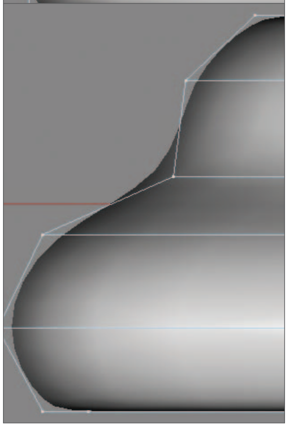
Switch to the front view. Select the top three rows of points for the entire object (including the newly extruded points) and choose the Scale Tool with the Y-axis locked (figure to right).

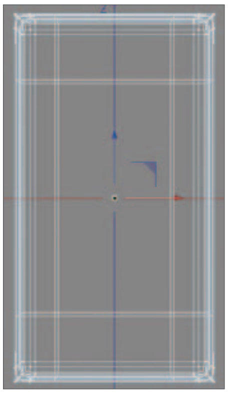




The figure at the left shows the results if we were to have done this scaling on a rectangular top. Since scaling is proportional, the ends would have shrunk more than the sides necessitating continual tweaking as we continue to scale. Later on we will simply move the ends away to give our table length.

It is now up to you to finish off the edge and attempt to match the profile to that pictured at the right. Notice that some of the rows have been repositioned vertically as well as being scaled. You can do this while the selection rectangle is still active (without switching to the *Move* Tool) by pulling the Y-axis arrow. If you wanted more perfection you could add other rows of points using the *Knife* tool along with *Shift* key. Now to lengthen our table.





From the top view, select all points in the top half. In the Coordinates Manager, enter 150 in the *Z Position* box and *Enter*.

The value of 150 will be added to the current Z position of each point. They will still retain their current relationship to one another. You can also divide and multiply numerical values within text boxes.

Move the bottom points down an equivalent amount (subtraction works also).

Comparing the two modeling methods

Either method would be appropriate for creating a table in a small scene with few other objects and not too many frames of animation. Following are some factors to take into account:

- The sweep and loft method gives you precise control over the appearance of the edge since the surface conforms exactly to the Spline. Positioning the points of a HyperNURBS mesh to get similar results is trickier and less direct.

- If you change the overall shape of the table by moving any end points of the edge path you will have to rebuild the top and bottom.

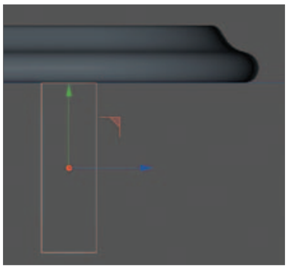
- If you want to change the level of subdivision with the sweep and loft method you must change the interpolation value for each of the Splines involved. With the HyperNURBS Object, only one value needs adjusting in the HyperNURBS.

- If you need to deform the object for any reason, as in your table coming to life or warping for some reason, you would definitely choose the HyperNURBS method. While it is possible to convert the sweep and loft into a single mesh which could then be deformed, those long, skinny triangles and long, undivided edge strips would present problems when animating.

- When the Sweep and Loft Object is converted into polygons for rendering, the polygon count is 3668. The cage used with the HyperNURBS Object contains only 102 polygons. The HyperNURBS will multiply this count during rendering, but you can easily control this with a single entry for *Subdivision Renderer*.

- What if you wanted to modify the curvature at the corners as seen from the top view? How would you have to proceed using each of the two methods?

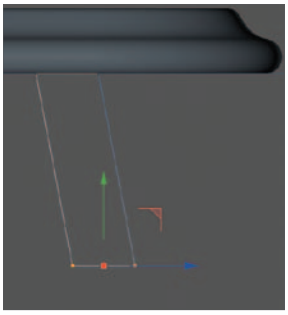
The rail (apron)



The rail gives rigidity to the top and helps strengthen the connection of the legs to the top. A Sweep NURBS is ideally suited for this. Since we will work with TableTop_2, let's store the others and clean up the Object Manager a bit.

Organize the Object Manager by storing all except TableTop_2 in a new Null Object named ..bk.. Hide ..bk. and move it to the bottom of the list. Move TableTop_2 up 10 units along the Y-axis.

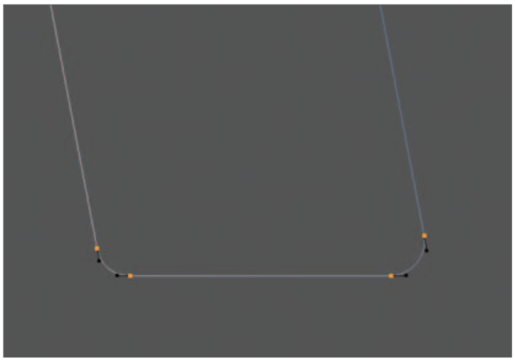
This move placed the bottom of the tabletop at Y = 0 to make positioning the rail and legs a bit easier.



From the top view, create a new rectangle in the XZ plane named railPath with a width of 280 and a height of 580.

This will serve as the sweep path for the rail. We can fine-tune its size later. It was automatically created in the XZ plane because you were using the top view. Next we will create the profile of the rail.

From the front view, create a rectangle named railProfile with a width of 12 and a height of 42. Move this over and position it about where it should be - Just under the top toward the right edge. Move in on the view.



It really does not matter where this profile is positioned, it will still be forced to follow the path. We have moved it here to better model it in relation to the top with the top. Simple tables have vertical rails. Expert crafts people can cut the complex miters necessary to angle the rails. To modify our profile we need to make it editable.

Make railProfile editable, choose the Point mode and select the two bottom points. Move them outward along the X until the rail is angled about 9 or 10°. Drag the bottom points until the bottom left point is vertically under the center of the top two points. Do not exaggerate the angle or your legs will splay out too far.

Now we will round those edges on the bottom.

With the two bottom points still selected, choose the Chamfer tool from the pop-up. Click anywhere in the editor and drag right.

Do not over-do the rounding. Add just enough to give a nice, smooth edge that will not hurt when you bump your leg into it. If you want greater control, you can set the value numerically in the Active Tool Manager. A value of 2 or 2.5 is just right (if you were to check the *Flat* check box you would get a true chamfer, which is a bevel). Time to take a look at our progress.

Create a new Sweep NURBS and name it Rail. Drag the two Splines into Rail in the proper order. Back the camera out to see your rail.

Unfortunately, it probably punches through the table. Remember, the sweep's position is determined both by the path, which is at Y=0, and also by the profile's origin within the profile. We could interactively drag our rail down the Y direction but the numbers are easy here so let's get it exact. Remember, we moved our top to have its bottom surface exactly at Y = 0. Since our rail is centered on the World origin, we simply need to move it down half its height.

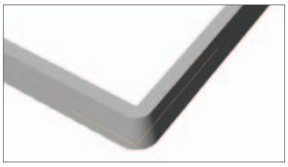
Make sure Rail is the active object and enter +21 for its Y value.

Hide the top and take a look at your rail from the perspective view.

Not quite right. There is at least a bit of rounding on all exposed surfaces, except the four corners of the rail, as seen from the top. Before reading on, think of how you can round the corners.

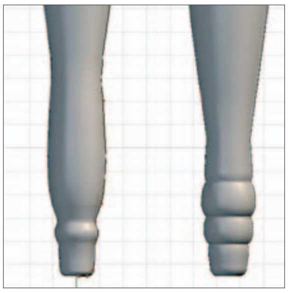
Round the corners!

If you do not like the look of the rounding, feel free to make modifications. Unlike drawing, there are no eraser marks.



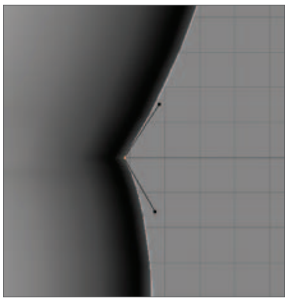
The legs

There are many ways to fabricate legs for a table. To be consistent in style, this table should have round legs turned on a lathe. These legs would most likely be squared off at the top so they could be most securely fastened. We could do this to our legs - but will you ever see this underside detail? If not, then a simple lathed NURBS will do. We will build and position one leg, then duplicate the others from it.



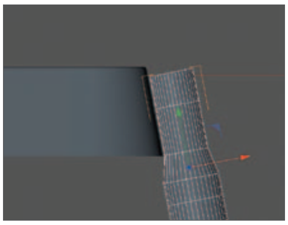
From the front view, create a profile (legProfile) of a leg about 250 units high. Once you have the approximate proportions, hide the top and the rail and fine tune your profile. When it looks similar to one of the two pictured previously, or you are satisfied, drop it into a Lathe NURBS Object named leg 1 (if you really like your spline, save your file!).

The Spline of the leg on the left uses 11 points (including one at the top center which is not absolutely necessary). To make the one at the right, use the *Shift* key while dragging tangent handles to get the v-shape between the rounded sections. Model your Spline in the front view only and use the Coordinates Manager to be sure the two end points are exactly on the Y-axis (X = 0) to completely close your object when lathed. Notice how the top portion is vertically straight so it will fit securely inside the rail (after rotation).

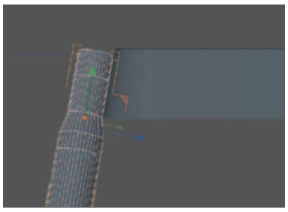


Make the rail visible, position leg 1 over to the right side of the rail so you can rotate it into position. Using the Rotation tool, rotate the bank axis until the top part of the leg is parallel to the rail as shown.

Enlarge the view for more accuracy while trying to match the angle by eye.



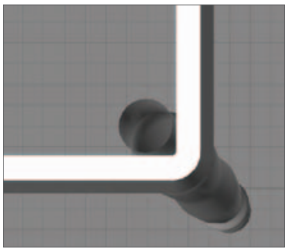
Now you need to rotate to match the angle from the side.

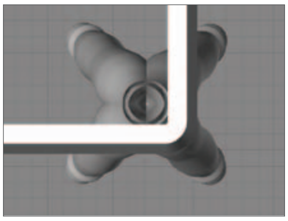


Switch to the side view. Move the leg to the left (object's front) until it is, again, just outside the rail. Rotate the pitch axis until the angle matches. Switch to the Move Tool and the World Coordinate System.

The object's axes have been rotated so they no longer point in the direction of the World axes. This switch will make it easier to position the leg.

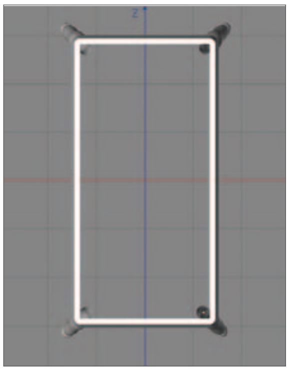
Adjust the height of the leg if necessary. From the top view, move the leg until it is properly positioned inside the rail.





Since our tabletop and rail are perfectly centered on the Y-axis, our other legs will appear in mirror positions of this leg. We can use this to our advantage. If we were going to "paint" our legs with some sort of uniform material, then Symmetry Objects would be our fastest route. However, we will use a wood shader on this table. Copying a single leg by symmetry would make the grain of all legs point in the same direction as the first leg. Since the legs themselves run in different directions, this would not look natural. The grain should run along the long axis of each leg. Therefore, we will make copies of the first leg which can be textured separately.

You could drag copies of the leg and then rotate and position each as you did the first. Use of the Coordinates Manager might help with this task. Here is a way to determine all the angles automatically:



Make sure you are in the top view with Leg 1 selected. From the main menu, choose Tools / Arrange Objects / Duplicate. In the Attribute Manager, set Copies to 3, check Generate Instances and click Apply.

The *Generate Instances* option produces instances that are automatically updated if you modify the Spline of the original leg (to which the instances "refer" for their shape). This is a distinct advantage in modeling flexibility but it will require just a tad more care when texturing. Now you need to position the copies. First we rotate the copies.

In the Coordinates Manager, change to the World Coordinate System (lower left text box).

We must rotate them around the World's Y-axis. If we rotate around the Object axis, their orientation will not be affected.

Also in the Coordinates Manager, change the heading of the first copy to 90, the second to 180 and you can figure out the heading of the third copy.

Now to move the copies. You can line them up by eye or take advantage of the fact they are symmetrically positioned around the World origin. For example, the leg that belongs at the lower left corner should end up the same negative X distance on the other side of the Z-axis as is its positive X value now. Therefore, if the X Position is, say 105.160, change the value to -105.160.

Position the legs by the numerical method for practice. If you position by eye, use the top view and make sure the *World Coordinate System* is selected.

Position the leg copies.

Drop your original leg into the Null Object containing the three copies. Name the Null Object Legs. Create a new Null Object, name it Table and drag the top (tableTop_2), the rail and the legs into it.

Unhide the top and arrange a good angle on your table in the perspective view. Render in the editor and admire your creation.

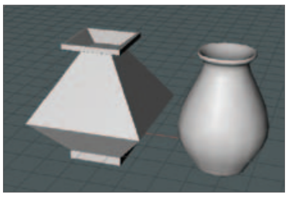
That's it. There are many variations on tables and many subtle refinements you could make to this one if you wanted. Later we will revisit this table to texture it and feature it in a scene that we will light and render. If you have time, you may want to create a fancier leg. Maybe the corners at the top are too rounded for your taste. This will require adjusting a few Splines, but should not present too much of a challenge. Here is a more serious challenge: Plan how you would model the leg if you needed to show the square top. Do not look at the hint in the fine print below until you have puzzled awhile. If You are ahead time-wise, do it! (*hint: eight sided lathe to a HyperNURBS)*.

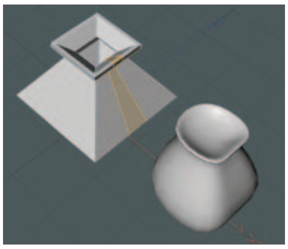
Pitcher Tutorial

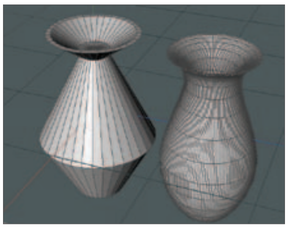
In this tutorial you will model a glass pitcher with an integrated handle and pouring spout using a polygonal model in a HyperNURBS Object.

This exercise will give you a taste of polygonal modeling in a situation where no other method will sensibly work. You could model the pitcher without the handle using a Loft NURBS, as you may have already done in a previously tutorial. You could then make a separate handle, but it would never attach smoothly enough for close inspection and would probably not produce satisfactory results when rendered as glass.

Choosing a technique for the mesh





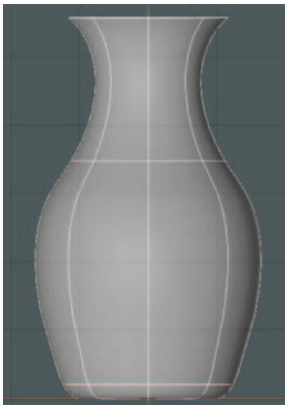


There are several choices for creating a polygonal mesh for a HyperNURBS Object. Probably the most commonly used initial object is the Cube Primitive; many animators even begin human heads with a simple Cube Primitive. Cubes begin with closed geometry, they are not capped with separate polygons like most other objects such as cylinders and SweepNURBS. Cubes also let you begin with a low polygon count and add points as needed. The figure at the right shows a HyperNURBS vase derived from a primitive using just the extrude and knife tools with some scaling of selected groups of points (the left-most of the figures shows a solid view of the mesh). This figure illustrates one of the most important points to consider in HyperNURBS modeling: Let the HyperNURBS Generator produce as much of the curvature as possible. The more points you add, the more difficult it will become to manipulate the geometry and maintain smooth curves, not to mention the added memory usage and increased render times.

One problem with the technique above, for our particular situation, is that the narrow handle would require the mesh to be knifed from the top view, causing the loss of our radial symmetry. Notice in the figure at the right how the roundness of our geometry is destroyed when the mesh is knifed to create a narrow enough region to construct our handle. We could move the new points out from the center to try to reestablish symmetry by eye, but this will be imprecise and difficult. As good as the cube is for creating meshes, it is not the right tool for this job. A Cylinder Primitive could also serve as a starting point for our mesh. Although this will result in many more polygons than our cube-derived mesh would, it will give us the narrow polygons needed for handle construction while preserving the radial symmetry. For the vase shown at the right, the cylinder was optimized after being made editable to get rid of the extra set of points at the top and bottom edges, where the caps met the sides. The points used to fatten the bottom and narrow the neck were done using the knife tool, as was done with the Cube Primitive above. The number of rotation segments was reduced to the minimum, which still allowed for handle construction. The cylinder would do the job but would require some tedious work to get the top edge smoothly rounded. Another problem with this approach is that, when scaling the knifed points in or out to produce the neck or belly of the pitcher, the thickness of the wall would change. This is not acceptable for a transparent glass pitcher and would be tedious to remedy.

You probably already know that a LatheNURBS Object might be an acceptable tool with which to begin our mesh. That is just what we will use, but it will also require some care in setting it up before converting it to a polygonal mesh. Let’s begin.

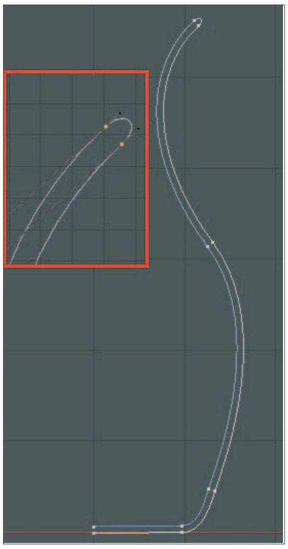
Creating the mesh



Create a new folder, "Glass Pitcher". Open and set up CINEMA 4D. Save the file as "Pitcher 1". Create a five point Spline as shown at the right. Drop the Spline into a latheNURBS and adjust the Spline's curvature.

Pay strict attention to proper lathe modeling techniques and make sure to begin at the origin. Make sure the shape is what you want from the very beginning.

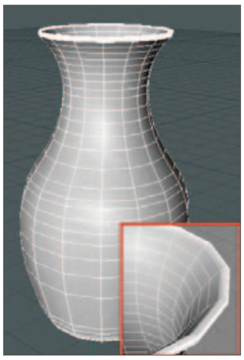
Use the Create Outline tool to create the inner surface and round the top edge (as done in a previous tutorial using Equal Tangent Length with some handle moving). Remember to uncheck Close Spline in the Attribute Manager.



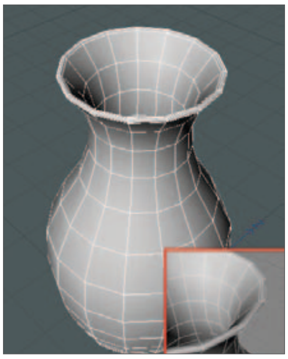
Again, make sure you are happy with the shape - soon you will be working with polygons.

Reactivate the latheNURBS and set the Subdivision to 15. Make sure you are in Gouraud Shading (lines) mode with Wire frame so you can monitor the subdivisions (that will end up as polygons in your cage). Study the mesh, especially the top edge (figure to left and below).

Why fifteen subdivisions? We want the handle in the center of the polygons but need the spout centered on a joint between polygons. Since these two features are opposite one another, we need an odd number of polygons going around our pitcher.



Close inspection of the wireframe shows far too many polygons at the top edge and where the sides meet the base. There is a 180° turn at the top edge where the outside rounds over to become the inside. Remember, the default interpolation method for Splines is Adaptive with Intermediate Points placed every five degrees. There are 36 points, hence 36 polygons, making that 180° bend at the edge. Since there are 15 segments around, this makes a total of 540 polygons just for the top edge. Let's arrange it so the HyperNURBS will do most of the work.



Change the Spline's interpolation method (*Intermediate Points*) to *Natural* and change the *Number* to 4. Take a look at the edge again as well as the entire shape (figure at the right).

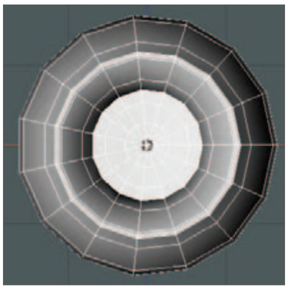
Much better - only five polygons going over the edge (we could have done this by using Adaptive spacing with the Angle set to 36 but then we would not have had enough subdivision along the sides to construct our handle and spout).

We are now ready to convert this object to polygons. This will make for a nice cage with a decent polygon count.

Make the latheNURBS editable, name it pitcherMesh **and drop it into a HyperNURBS named Pitcher.**

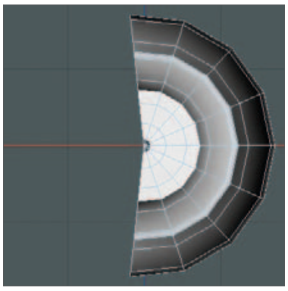
The faceted look resulting from only fifteen rotation segments has been eliminated by the HyperNURBS Object. That top edge is looking pretty smooth as well.

Making the spout



Because of careful planning and good design we will not need any new points for our spout. We will do most of our polygon selecting from the front of the spout, which happens to face to the right. Since we also need to select the inside surface, we need to uncheck *Only Select Visible Elements*. However, how do we avoid selecting the polygons on the other side?

Deactivate the HyperNURBS and switch to the top view.

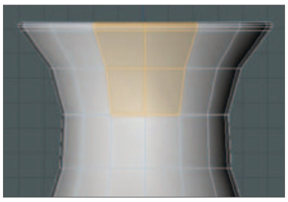


Note how the right side has the polygon edges right on the axis - this is the spout side. The left side has the polygons centered on the axis, perfect for the handle.

Make sure Backface Culling is unchecked. Using the rectangle selection tool with Only Select Visible Elements unchecked, select the entire left side of the pitcher. In the main menu choose Select / Hide Selected.

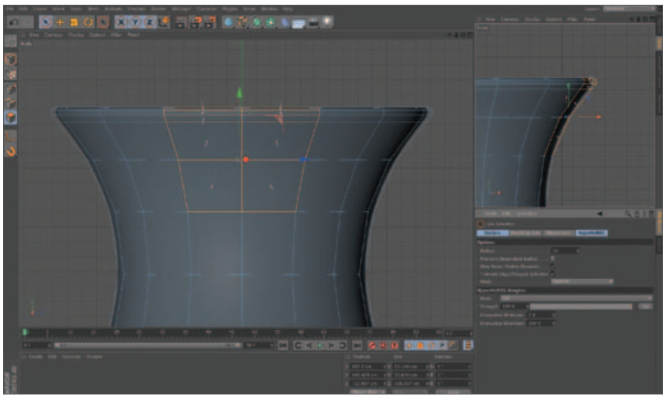
This will keep any polygons on the left from accidentally being selected and manipulated.

Editing points is often faster with HyperNURBS off. However, you also need to keep an eye on what the smoothed mesh looks like. You can easily toggle the HyperNURBS on and off while modeling using the *Q* key (try it). You may also want to try the *Isoline Editing* mode (in the Options menu of the view panel). This mode projects the cage onto the smoothed HyperNURBS Object.



From the side view, select the polygons shown at the right above. Create a New View Panel, dock it to the right of the Object Manager and set it to the front view (the pitcher's side) with Gouraud Shading. Stretch the window to the left to make it larger and move in on the spout area (lower right).

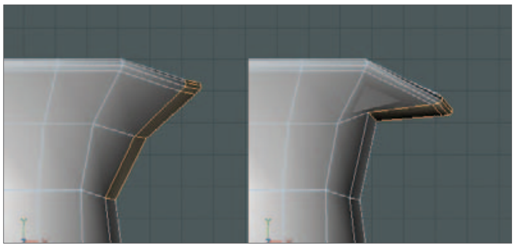
This will allow you select from the spout's front in the main view and shape the spout from the side in the new viewport.



Rotate the selected polygons a bit around the bank axis only (near right) then, using the Move Tool, lower them down as shown (far right).

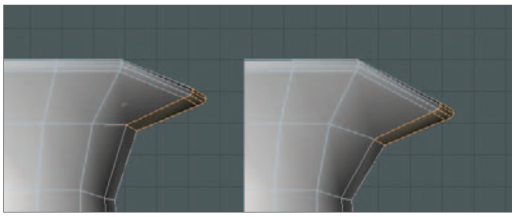
These operations would work just as well in point mode but it is a bit easier to select the proper polygons.

Deselect the bottom row of polygons. Rotate (near right) and move (far right) the polygons again to make the spout more pronounced.



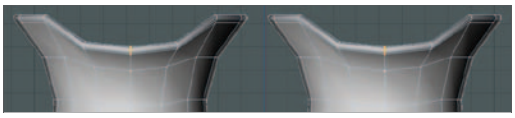
Now we need to work on the center row of points to make the channel a bit deeper in the middle of the spout.

Select the points as shown (near right). Rotate them around their bank axis just a bit and lower them a bit. Deselect the points on the bottom (far right) then rotate and lower down subtly.



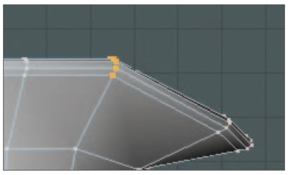
Your spout should now appear similar to the one pictured at right.

Select the points shown at the far right. Lower these points, as shown above.

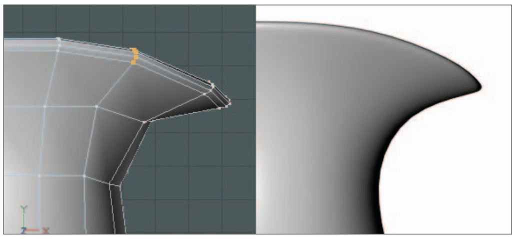


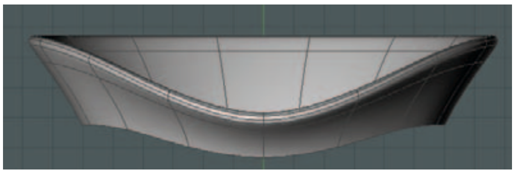
Your spout is finished from the side view and should look like those at the right (rendered with the HyperNURBS turned on to the far right). We will finish the spout with a little tweaking from the top to slightly narrow its width.

Be sure the cage is selected in the Object Manager and the Polygon mode is active. From the main menu, choose Select / Unhide All.



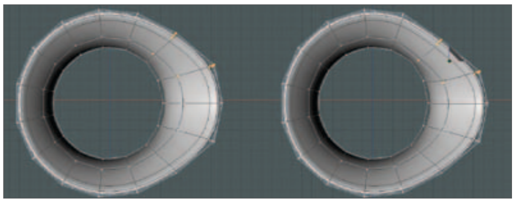
You can also hide and unhide points when the Point mode is selected.



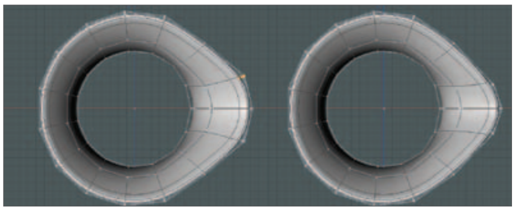


From the side view, select only the polygons shown at the top right and use Select / Hide Unselected to hide the rest (already hidden in the image shown). Switch to the top view and select the points shown (near right) (only Select Visible Elements should still be unchecked). Move these points in along the Z-axis as shown (far right). Select the symmetrical points on the other side and move in the same amount.

You could select the points on both sides together and scale inwards along the Z-axis but this would scale the relationship between the points defining the top edge making it appear flat.

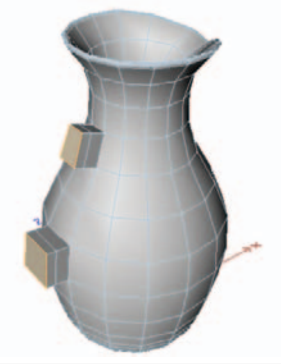


Select the points shown at the near right and also move inward along the Z-axis a little. Repeat for the other side.



Your spout should now look like that at the far right and like the rendered version at the left. Look at your spout with the HyperNURBS Object active from different perspective angles. Make further tweaks if you wish using the tools and techniques just practiced. Maintain symmetry as you adjust and remember to select the inner surface points as well as those on the outside

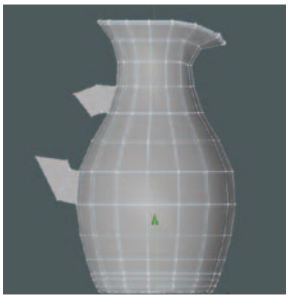
Making the handle



The spout gave you good practice in the manipulation of points and polygons but you could have constructed the pitcher to this point using a lofted spline. In the making of the handle you will get a true taste of polygon modeling.

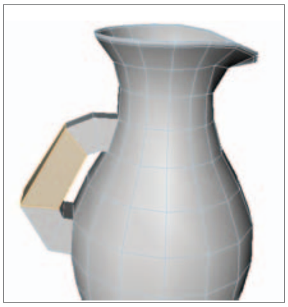
Unhide all polygons if you haven't already done so. Working with the side directly opposite the spout, select the two polygons shown (to the right) and extrude them twice.

You can select these with the Move Tool, which will not select the inner surface polygons. If you use the selection tool, make sure you enable *Only Select Visible Elements*. The exact amount of extrusion is not important, you will move these points into position manually.



Switch to the front view and select the Point mode. Make sure Only Select Visible Elements is now unchecked. Select the points in pairs (the front, visible point plus its symmetrical hidden partner behind it) and move them as pictured to the right.

Again, the precise position is not critical. The main goal is to get the two outer-most extruded polygons to somewhat face each other so we can bridge between them.

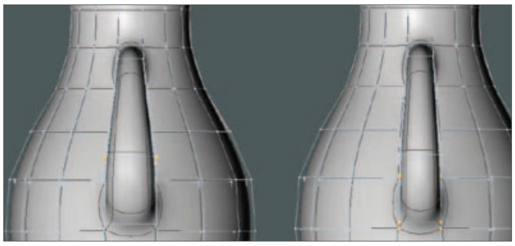


Select the two facing polygons and bridge them. Reactivate the HyperNURBS and, working from the front view, move the points until your handle is looking good (far right).

Because the top poly you extruded was narrower than the bottom poly, the handle is uneven in width.

Keep the main viewport in the front view. In the other viewport choose the side view. Then, from the Cameras menu, select Left. Select the four pairs of points making up the bottom of the handle and, using the view of the left side, scale the points in along the Z-axis to make the handle bottom more narrow (to the right is a before and after shot).

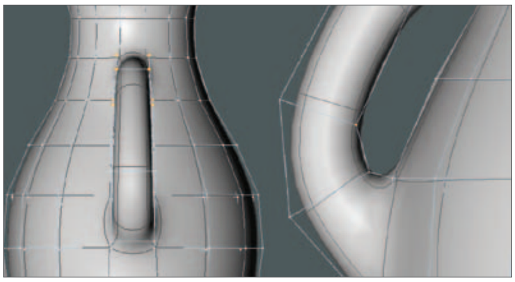
Do not make it too narrow - you are also going to widen it a bit at the top.



Select the four pairs of points forming the top of the handle and the two pairs where the handle connects with the side of the pitcher.

Be very careful selecting these last four points. Make sure you do not select any points on the inside. This is best done using the *Live Selection* Tool while rotating the model around in the perspective view with *Only Select Visible Elements* checked.

Scale these points outward along the Z-axis until your handle is relatively uniform (near right). For a final touch, drag the two points at the bottom of the handle, where it meshes with the side, down a bit to give the handle a smoother, molded look.



Now we have a pitcher. Hope you were successful! This probably seemed quite laborious to you and took between one and two hours to complete. However, as you practice and develop skills with polygonal modeling, you will become consistently faster. Polygonal modeling, along with the use of HyperNURBS, really is the best way to make most objects beyond lamps and chess pieces and the only sane way to make organic shapes such as characters. Polygonal objects also offer advantages in texturing and animation although Spline Generators have their advantages in certain situations.

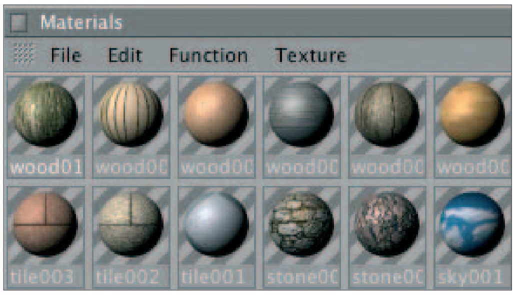
Even as tedious as some aspects can be, polygon modeling can be satisfying not only with regard to the results but for the creativity involved in solving the problems of how to create and efficiently manipulate a polygonal mesh for a particular situation.

Introduction to Materials

Overview

Materials are applied to surfaces to assign them a wide variety of properties. This includes color, roughness, how the surface reflects light, transparency and several other parameters which, all taken together, comprise a material. There are several technical terms associated with the process of creating and applying textures:

A **material** is the general term for all properties - all the different parameters affecting the surface that are stored at a single location. That "location" is a called a material and is represented by an icon consisting of a name and a sphere showing a preview of the material. Materials are stored and accessed from a window called the Material Manager. Materials can be applied to objects by dragging their icons from the Material Manager onto the object's name in the Object Manager, or directly onto the object in the viewport. This is called **texturing** an object. Once the material has been applied to a specific object, it is referred to as a **texture,** but the term "texture" also has other, more specific meanings.

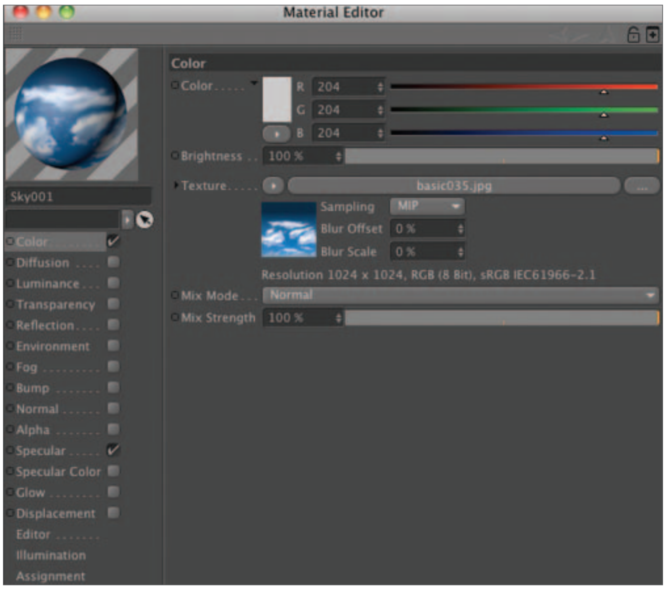


The term texture also refers to patterns and images that are used within materials. When a bitmap image (one composed of tiny pixels which make up the image on your screen, a TV or in a digital photo) is used, it is called a **texture map**. Texture maps may consist of smudged spots to make a surface look dirty or used, painted patterns depicting snake skin, an actual photograph to be used as a picture in a frame, or even a movie.

Textures may also be generated mathematically with user-defined parameters, just as with parametric objects. Textures generated and controlled in this way are called **shaders**. Some shaders produce a 2D texture which is then mapped ("wrapped") onto the surface of the object. The most powerful shaders are volumetric, or 3D, shaders. These are not mapped onto the surface as with the 2D shader but are calculated based on the 3D volume of the object. This is perfect for such textures as wood. If you were to cut a chunk out of an object textured with a 3D shader meant to produce wood grain, the wood's texture would appear correct, just like cutting into a piece of real wood. 3D shaders are even more useful in that they can be applied to the surface without the distortion inherent in trying to stretch a flat 2D image onto a curved 3D surface.

Materials

Materials are stored in and accessed from the **Material Manager**. You can create new materials here from scratch or *Control+drag* existing materials to make copies which can then be modified. Materials can be dragged onto objects to texture them. The material itself will remain in the manager and can be used as many times as you like. Any changes you make to the material will be applied to all objects textured with that material.



Sky001

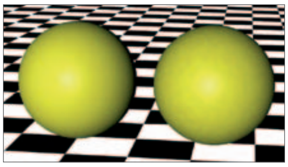
Color ☑
Diffusion ☐
Luminance ☐
Transparency ☐
Reflection ☐
Environment ☐
Fog ☐
Bump ☐
Normal ☐
Alpha ☐
Specular ☑
Specular Color ☐
Glow ☐
Displacement ☐
Editor
Illumination
Assignment

Color
Color R 204
G 204
B 204
Brightness 100 %
Texture basic035.jpg
Sampling MIP
Blur Offset 0 %
Blur Scale 0 %
Resolution 1024 x 1024, RGB (8 Bit), sRGB IEC61966-2.1
Mix Mode Normal
Mix Strength 100 %

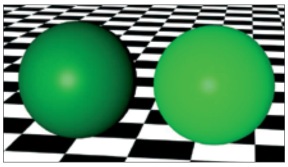
Double-clicking on a material will open the **Material Editor** window where you set all the parameters defining how the material will appear. The editor has a list of **channels**. Each channel controls a certain aspect of the material such as its color. A check box lets you define which channels will be active for that material. A click on each channel's name or check box gives you access to its parameters. You may render in the editor at any time to see the results of any changes made. Materials may also be edited in the Attribute Manager, but the Material Editor is better suited to working with materials.

Channels

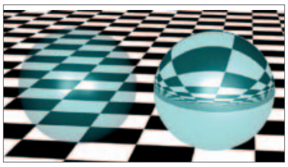
Color: As the name suggests, this channel applies color to the surface. However, you are not limited to one color to be evenly applied to the entire surface. Through the use of textures, colored patterns and images may be used. This may include photographs, images from a paint or drawing program, or patterns calculated using mathematical formulae (shaders). As with many of the other channels, you may blend a texture with a solid color using the *Mix* parameter.



Diffusion: Uniformly colored surfaces generally do not look realistic. Real objects usually show some slight variation in color. In older objects, for example, this may be wear and tear or dirt. If you apply the diffusion using the *Diffusion Level* slider at the top, the entire surface will have its underlying color or texture muted in appearance. A more useful method of applying diffusion is by using a texture map or a shader such as *Noise*. This will cause the diffusion to be applied in a pattern, with some areas showing the diffusion and others remaining unchanged (notice the slight bit of unevenness in the coloration of the right sphere above. This is due to noise used in the diffusion channel with its strength set to 10%).



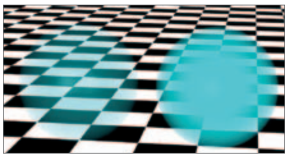
Luminance: Luminance refers to light emitted by the object independently of any external lights. This is useful for such effects as a TV screen when used with a texture or movie. Sometimes a slight amount of luminance, using the same settings as in the color channel, can be used to reduce the amount of contrast on a particular object under harsh lighting.



Transparency: The amount of transparency can be adjusted using the *Brightness* slider. A value of 100% (white) makes the object completely transparent, a value of 0% (black) will make the object opaque. Most transparent objects will bend light as it hits the object. This is called **refraction,** and the amount of bending is dependent upon the characteristics of the object's surface (at the right, the second sphere has refraction enabled). You can adjust this parameter by entering a value for n in the text box under *Refraction*. Values for common substances such as glass (n = 1.5) or water (n = 1.3) can be found in the help files. Textures can be also be used in the transparency channel to produce special effects such as a movie projector or sunlight shining through blinds.

Reflection: This will cause your object to reflect other objects on its surface. Adjust the amount of reflection using the *Brightness* slider.

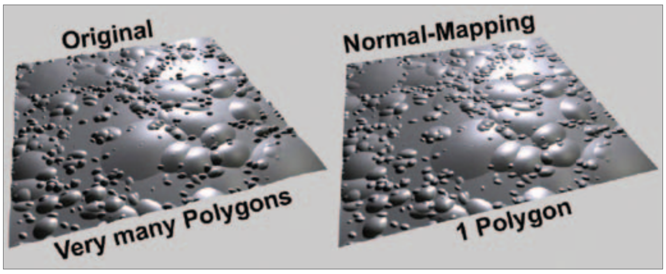
Environment: Reflection often helps us recognize the composition of objects in the real world such as the characteristic shine of metals. It is often easier to produce this effect, especially if there are few objects in the scene, by providing some general pattern to be reflected. A cloudy sky, for example, is sometimes used as an environment map to help give metals their reflective quality (both spheres above show reflection. The one at the right also uses an environment map to cause a cloudy sky to be reflected as well).



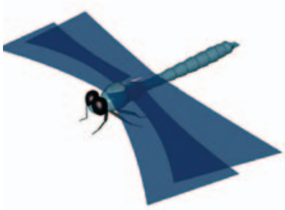
Fog: This channel is useful if you wish to create a specific type of fog. The material can, for example, be applied to a low, flat, rectangular object to simulate ground mist (a global fog is also available, which controlled through an *Environment* object and can be used to place fog throughout an entire scene).

Bump: This is one of the most important channels; most objects are not perfectly smooth. Besides the obvious dimples in a golf ball and texture of human skin, bump could be used to model the face of a coin with a fraction of the memory and modeling time compared to actually modeling a presidents head with polygons. However, bump does not affect the edges, or silhouette, of an object. It only affects the surface (just like the *Phong* Tag). Bump requires a texture map or procedural shader of some sort.

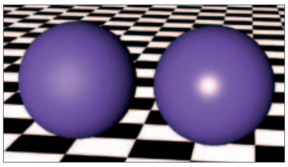
Normal: This material channel is for use with Normal textures via Normal Mapping. What on earth is Normal Mapping? Well, Normal mapping is a technique that comes from computer game development that makes it possible to give a low-polygon object (a low level of detail) a detailed, seemingly structured surface, resulting in low render times. Bump mapping uses a grayscale bitmap to generate height data. Normal mapping uses an RGB texture with coded Normal orientation (the Normals can be oriented in various directions).



Alpha: An alpha channel contains information in a texture map or shader about where an object should exist. This means the alpha channel can be use to further "sculpt" your object after modeling. When active, this channel will cause any places where the map is black to be seen, showing whatever settings you have made in the other channels of your material. However, where the map is white, the material will totally disappear. The dragonfly's wings are defined by textures on simple, flat objects (bottom right) (the wings are actually simple rectangles that have been deformed). The final shape of the wings is due to the alpha channel. This method of modeling with textures is also indispensable with leaves and can even be used for entire trees, buildings and much more if they remain in the background and the camera movement is not too severe.

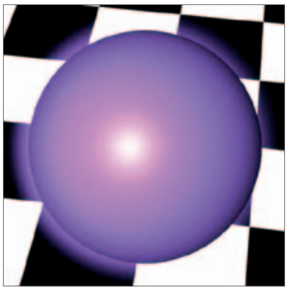


A more important use of alpha channels is to layer materials. The places where the top textures are invisible allow the underlying textures to show through. This makes it possible to build up complicated multi-material surfaces.



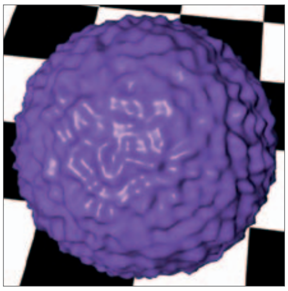
Specular: Whereas only a few surfaces exhibit the type of reflection which forms images of other objects, most surfaces reflect light to some degree. Some highlights are dull and spread out, like paper, others are sharp and intense such as metal. These characteristics are adjustable as the *Width* and *Height* parameters in the specular channel.

Specular color: Sometimes, especially with metals, the specular highlights can be of a different color than the metal or the light. This can be adjusted here.



Glow: The glow channel can be used to create a halo-like glow around objects (*outer Strength*) and over the surface (*inner Strength*) of the object. The glow can be animated using the *Frequency* and *Random* parameters. Glow is useful for such effects as neon lights and lightening but is limited by the fact that objects passing in front of a glowing object will have the glow bleed over their surface in an unrealistic manner.

Displacement: This is similar to bump but the surface geometry is actually distorted. This means the edges as well as the silhouette of the object are affected. How fine or rough the effect is depends on the amount of subdivision in your object. Nice effects require high polygon counts and longer render times.

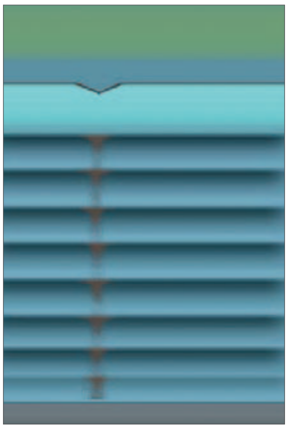


Illumination: Here is where you control how an object with the material behaves with advanced rendering features. *Global Illumination*, also known as radiosity, affects how a surface reflects diffuse light. If sunlight were to shine through the window of an otherwise darkened room the room would be illuminated in places where the sun's rays didn't directly shine as a result of light rays bouncing off various surfaces, such as the walls and floor. Caustics allow light to pass through transparent objects or reflect off shiny surfaces and affect other objects.

Textures

Except for the simplest of objects, the most important objects in a scene will contain materials having textures in one or more of their channels. Remember, the term texture may refer to a material after it is applied to an object or to images which give materials some sort of pattern or graphic other than a solid color. Some textures are bitmapped images from paint programs or digital photographs. Others are procedural textures (shaders) which use adjustable parameters to generate the patterns mathematically. Some shaders have their own separate editor. Textures give surfaces unique characteristics, by either varying the color, surface consistency or even the shape or when used in the alpha and displacement channels. We will talk about different texture types in this section and discuss how they are placed onto surfaces (mapped) in the next section.

Texture map (bitmapped texture)

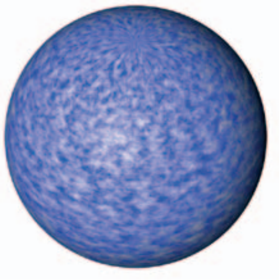


A bitmap refers to a "paint" image, one where the image is composed of rows and columns of pixels, which is also how the image on your computer or TV screen is created. To the right is the bitmap that colors the body of the dragonfly pictured earlier. The lower image is an enlargement showing the individual pixels that make up the image. Since the information in the image is fixed into these pixels when created, the resolution of bitmaps remains the same. However, as these bitmaps are re-sized to fit the object on the screen, the computer interpolates (calculates new pixels) when necessary. Some of these methods, especially the default *MIP* mode, do a pretty good job stretching or shrinking the texture map as needed.

There are many sources of texture maps. You may use your own photos or bitmapped images from other sources, such as the net (make sure there are no copyright or plagiarism issues!). You may also modify existing images or create images from scratch using a program such as Photoshop or even CINEMA 4D itself. Larger bitmaps offer more detail and look better in close-ups. The downside is increased memory usage and longer redraw and render times.

When texture maps are applied to objects they are often tiled. This means they are used at a size less than the full object dimensions and repeated in rows (X) and columns (y) to cover the entire surface. This can present a problem with seams unless your texture has been designed to be seamless (you can easily see how the sphere to the right has been tiled, especially in the Y direction).

Shaders

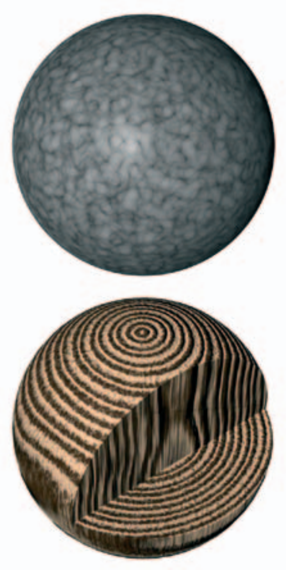


Shaders (procedural textures) solve some of the problems associated with bitmapped textures, such as seams, when tiling. You will notice in the figure at the right that no seams are visible, in contrast to the bit mapped image shown above. However, you can still see the texture being pinched together toward the top of the sphere due to the need to map a two-dimensional onto the three-dimensional surface using some sort of projection (spherical in this case).

Shaders are parametric. Each shader offers different parameters whose values can be manually adjusted. The values are input into the internal formulae which then calculate the texture during rendering. Some standard applications for shaders are bricks, marble, water surfaces and clouds. Some shaders can also be animated including flames, swirling galaxies, fog and water surfaces.

Many shaders can be used in material channels just like bitmaps. These can be accessed in the relevant channel in the Material Editor by clicking on the triangle after the *Texture* text box. Other shaders are self contained having their own editor window and adjustable parameters. These can be found under the Create menu in the Material Manager. Shaders are excellent for a wide variety of surfaces ranging from geometric patterns such as tiled floors to more irregular patterns like the skin of an alien creature. With a bit of creativity, shaders can be used to create fin patterns on a fish, louvers in window blinds, a sea floor bottom and much, much more. The gradient of color in a sunset sky can be easily produced with shaders (the sky's not even the limit!).

3D Shaders



3D shaders, also called volume or volumetric shaders, have a particular advantage over other shaders. When a 2D shader or texture map is applied to a surface it must be deformed from its flat state onto the curves making up the surface of the object. Imagine wrapping a poster around a beach ball. When this "wrapping" is done, a certain amount of distortion is inevitable.

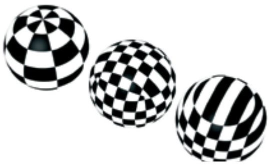
A 3D shader is not mapped onto the surface but calculated based on the volume of the object as if it were part of a solid, i.e. as if a texture were already present in three-dimensional space and the surface of the object then slices off and discards the portion outside the boundaries of the object. This results in a surface that appears to be a natural extension of the interior of the object. Notice in the figure at the right, there is not only an absence of seams but no pinching of the texture takes place, either. The texture appears uniform over the entire surface. The best example is wood. The growth rings and grain of the wood pattern extend in a particular direction independently of the shape of the object. If the object were to be sliced, the grain would appear correctly, just as if cutting into a real piece of wood.

Another feature of most shaders is that they are resolution independent. This means that no matter how much you scale the shader, CINEMA 4D calculates all the detail needed at that size. On the other hand, a bitmapped image has a fixed resolution. When it is enlarged past a certain point, you will begin to notice a loss of detail, even though such techniques as MIP interpolation do an admirable job at filling in when the texture map is stretched.

Volumetric shaders are ideal for natural materials that have an outer surface that is a continuation of the inside composition, like wood or marble. However, 3D shaders are also used to solve the texture mapping problems mentioned above. For example, organic objects frequently use some sort of fine bump pattern. When created as a 2D shader, the bump generally shows seams, puckering or regions where the scale of the bumps is different. The *Noise 3D* shader, used in the bump channel, eliminates these problems and results in a consistent surface.

Texture mapping

Bitmaps and 2D shaders must be "mapped" onto a surface when a material is applied during texturing. This process is called projection. There are a number of different modes of projection that are accessible in the Attribute Manager when the *Texture Tag* of the object is selected (a Texture Tag is created when you first apply the material to the object). Additional parameters involving the placement of textures are the position of the texture relative to a reference position on the surface of the object (the *Offset*), and the relative size and rotation of the texture. You may also choose whether to have the texture keep repeating itself (tiling) if you have set its size to less than that needed to cover the entire surface.



As mentioned before, wrapping a 2D, flat texture onto a 3D surface always presents problems. Think about the maps (2D) you have seen of the Earth (3D). In the most common geographic mapping technique, Antarctica looks disproportionally large. This is because it is stretched to make it the same width as the equator when actually, right at the South Pole, all the dimensions converge at a single point. If you were to apply a photograph to a sphere in the same manner it would appear pinched together at the poles. If you wanted to reduce the distortion at the poles, you would have to distort your texture map during its construction in much the same way as a geographer does when making a 2D rectangular map of the Earth. This projection type is called *Spherical* mapping. It is useful in many instances beyond just spheres, especially when the deformed ends can be placed in unimportant or hidden areas of the object.

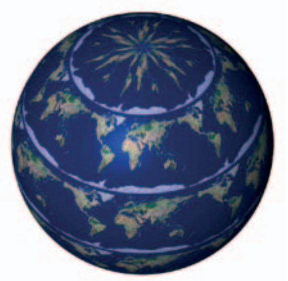
The figure at the right shows spherical, cubic and flat projections of a checkerboard shader onto a sphere. The *Cubic* projection type will solve the problem of distortion at the poles, or ends, cabinet, these seams will naturally fall on edges and not be noticeable.

Flat surfaces such as floors and walls are easy, this is what the Flat projection is perfect for. There are a few other projection methods you can read about in the help files.

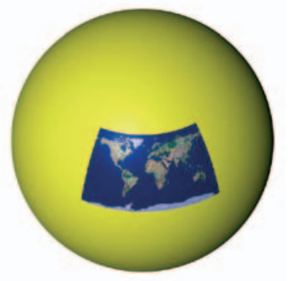
UVW mapping

One special projection type, *UVW Mapping*, deserves special attention. First of all, this is not a unique projection method, it uses one of the other projection types as a base. The symbols U, V and W refer to orientations of textures on surfaces in the same way X, Y and Z refer to spatial orientations of the object itself. UVW Mapping locks the texture onto the surface. If the object is subsequently deformed, the texture will be deformed along with it instead of slipping across the surface. The classic example of UVW Mapping in action is making a waving flag look realistic. This mapping mode will keep lipstick on lips during a smile and prevent skin from crawling during an arm movement. UVW Mapping is necessary anytime you apply a material with textural detail to an object that will be deformed either during animation or in further modeling stages.

Tiling



When a material with a texture map is first applied to an object, the default is to stretch the texture map so it completely covers the surface once. You may make the scale of this mapping smaller and have the map repeat itself on the surface as many times as you wish either in a row (the U direction) or in a column (the V direction). Our Earth, at right, has been tiled six times in both directions. The number of tiles is determined by the value in the input box under *Tiles* or the values of *Length* which is the percentage of the surface each tile occupies.



If you disable the Tile box, a reduced scale texture map will cover only a portion of your object. Additional underlying textures will then show through the untiled area (figure at the right). This, along with other techniques, is useful for building complex surfaces as well as the more common application of putting labels on bottles.

The tiling, length and offset parameters will not work with most 3D shaders. They generally have their own input boxes to serve the same functions. With these shaders you can achieve the same function as *Offset* using the *Position* input fields in the Coord Tab of the Attribute Manager. Likewise, the *Size* fields serve in the same capacity as *Length* does with texture maps.

An even more precise method of applying a texture map to a specific region of an object is through the use of the Restrict to Selection field. There are ways to select groupings of polygons and name these selection sets. The name of one of these selection sets may be entered in the Attribute Manager to keep your texture map restricted to just that portion of the object.

Texturing is one of the most difficult, time consuming and yet important tasks in the production of a quality rendering. Pencilman's surface, for example, is textured with several materials. In the screenshot of the Object Manager above, you can see seven Texture Tags plus the tags for selection sets that allow the textures to be applied to specific areas. You should read more about the various features and examples of how textures are applied in the Help files. The tutorial will take you through the introductory concepts but much practice will be needed to master the art.

MAXON CINEMA 4D also includes all the functions of BodyPaint 3D R3, which solves many of the problems associated with creating the type of non-uniform texture maps characteristic of complicated surfaces such as human faces. CINEMA 4D therefore lets you paint directly onto the 3D surfaces of objects. However, the quality of your paint job is dependent on how the UV mesh is mapped from the surface polygons. The adjustment of this mapping for something like a face requires practice, patience and time.

Study the surfaces of everyday objects. Beyond just the dominant color, note how the surface interacts with light. Are the specular highlights broad and dim, sharp and bright or some other combination? What kind of fine surface texture does the object have? How does the texture appear at the distance you might view this object in a scene? Is the color uniform, mottled or have some other pattern of non-uniformity? Begin to train your eye now so your scenes will be more realistic and believable in the future.

Materials Intro Tutorial

This tutorial follows the Introduction to Materials discussion describing various features of materials and how they are applied to objects. The CINEMA 4D help files should be frequently consulted as you proceed through the unit on materials. The help files contain many more details and examples of usage.

1. Materials intro (color and specular)

Let's begin by creating a basic material and applying it to a simple object. We can then progressively add channels to the material and view the results in the editor.

Open CINEMA 4D and make any standard adjustments; make sure you are in the perspective view. Save the file as "Materials Tutorial" in your "Intro Tutorials" folder and continue to save as you proceed through the tutorial. Create a sphere primitive and move the camera in until it occupies about 75% of the view. Next, create a new material by clicking anywhere in the Material Manager and use the keyboard shortcut Cmd+N.

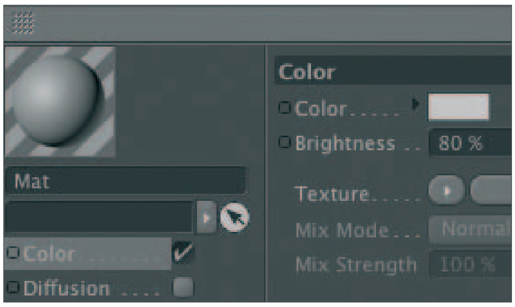
Cmd+N is the shortcut for creating a new element. In a program like CINEMA 4D, which has lots of different windows, the element created depends upon which window is currently active. A click in the Material Manager will make it the active window and pressing *Cmd+N* will create a new material. If the editor (view window) is the active window, then *Cmd+N* will create a new CINEMA 4D file.

Open the Material Editor by double-clicking on the icon of the new material.

Standard materials can also be edited in the Attribute Manager. Use the Attribute Manager if it is easier for a certain task. However, the Material Editor is better organized and faster for all but the most basic materials. By default, all new materials are created with the *Color* and *Specular* channels active. We will first change the color from the gray default color.

Click on the text "Color" in the list of channels on the left side of the Material Editor.

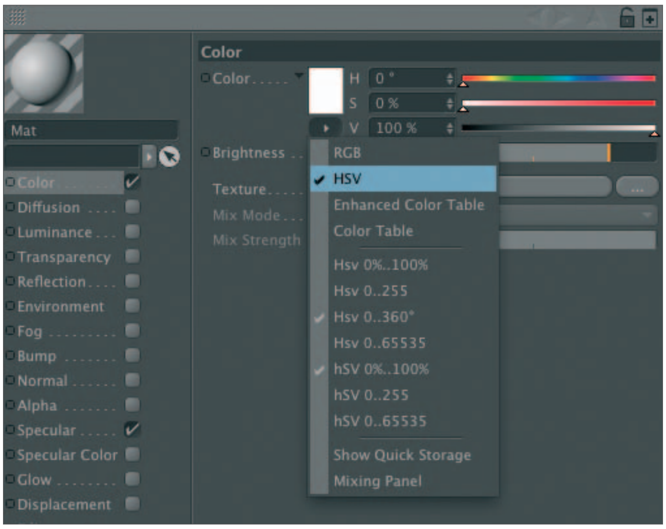
This channel was probably already selected by default. The selected channel appears with a highlight box around the text (do not click the check box unless you wish to make the color channel inactive). To the right of the list of channels is the "panel" which offers access to the channel's parameters. Under the panel's name ("Color" in this case) the word "Color" appears again with a small triangular arrowhead to its right and a color sample to the right of that.



Brightness . . 80 %
Mix Mode . . . Normal
Mix Strength 100 %

If the arrow points to the right (and sample appears small), click on the arrow (figure previous page) to reveal the color selection parameters.

The color sample should now be larger, with some sliders to the right. The standard color system used for computers and video is RGB (Red/Green/Blue). This means all colors in this system are made from a combination of these three colors. We will be using a different system that is easier to visualize, especially if you are new to graphics.

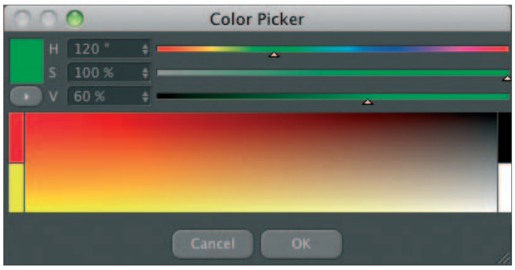


H 0 °
S 0 %
V 100 %

Click and hold on the arrow just beneath the color sample box. Drag down and select HSV (right).

In the HSV system, **h** is hue, which is the base color. Think of the colors as spread out in a rainbow pattern around a circular color wheel (right). The hue angle is some degree from 0° to 360° representing the shade of color as you go around the color wheel. Notice, as you go counterclockwise from red (at 0°), that the colors are ordered like those of the rainbow.

The saturation value **S** is a percentage specifying how intense the color is: Lower values result in a weaker color. For instance, pink is a less saturated version of red. The value **V** specifies how dark the color is to be. This is a percentage ranging from full color (100%) to black (0%).



H 120°
S 100 %
V 60 %

A click on the color sample may bring up your operating system's color picker. Unfortunately for Macintosh users, newer versions of the color picker do not show numerical values along with the HSV color wheel (shown at the right is an older version). Therefore, we will us CINEMA 4D's color tools. The hue values are shown along a line (slider) but you will notice red at both ends. Just imagine the slider curving back to close in a complete circle and you have the color wheel shown at the right.

There are several different color pickers available. We will predominately use the HSV picker in this course.

In the HSV color picker, change the Hue Angle value to 120, the Saturation to 100 and the Value to 60.

Note how the changes have been applied to the sphere in the preview box of the Material Editor.

Apply the material to your sphere by dragging the material's icon from the Material Manager onto the object's name in the Object Manager.

Notice the Attribute Manager has changed to show the Texture Tag's properties. This defines how the material will be applied to this particular object.

In the Attribute Manager, change the Projection type to Spherical. Leave the other parameters unchanged and view your textured object in the editor window.

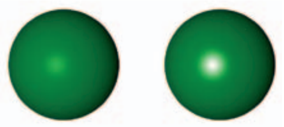
Since it is easy to accumulate a number of materials in the course of a project, it is a good idea to give them unique names. Although you can name materials by double-clicking on their name in the Material Manager, make the name change in the far upper left text box as long as the Material Editor is open.

In the text box at the top of the channel list in the Material Editor, name the material "green". After performing some other mouse action, you will note the name change under the material's icon.

In this case we named the material based on its color - green. In most cases it is desirable to give a material a name related to the object it will texture, for example "wall". This makes it easier to track down a material for later modification.

Now we will change the way an object with this material will display the bright highlights of lights striking the surface. This is called specular reflection, or just *Specular* . This is the reflection of light itself, not of images of other objects as would happen in glass, chrome or a mirror.

Activate the Specular channel and change the value of Height to 100.



Notice how the highlight on the preview sphere in the upper left of the Material Editor has become brighter. You will also see the change updated on any objects in the viewport to which the material is applied. However, the effect may appear rather crude because your sphere is not highly subdivided.

Render in the editor (*render Active View*).

The specular highlight on the rendered version should appear quite smooth. This is the work of the *Phong* Tag which smooths the texture. Check the preview in the Material Editor for a mini rendered version and render in the viewport when necessary to check your progress during the texturing phase.

2. Texture projection types

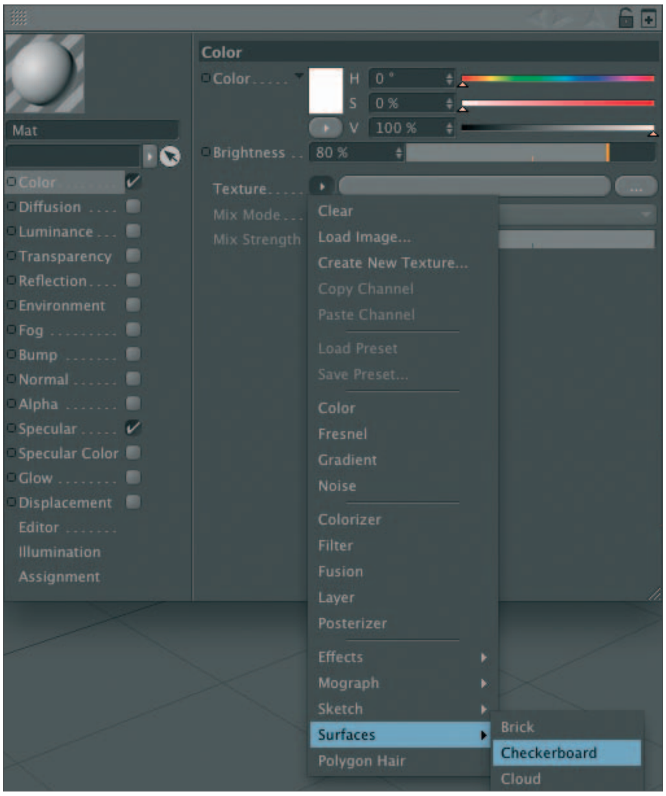
Before we investigate other channels, we will take a look at the different projection types since most of the other channels often use texture maps.

Select the Color channel of your "green" material.

In the color panel of the Material Editor you will find the "text" Texture with an arrowhead to the right. A click on the arrow (right) will bring up a menu with various texturing options. From here you have the option of loading texture maps saved from other sources (or previously rendered CINEMA 4D files) or selecting from many different shaders and functions which modify shaders.

From the *Texture* menu, select *Surfaces / Checkerboard*.

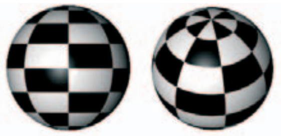
We will tile the texture to more clearly see the results of projection types.



Select the Texture Tag associated with Sphere in the Object Manager. In the Attribute Manager, change the *Tiles X* to 4. Use the *Tab* key twice to move to the *Tiles Y* box and enter 4 followed by *Enter* or *Return*.

Observe the changes to the sphere. The checkerboard looks pretty rectangular around the equator but pinched at the poles. This is characteristic of spherical projection. You will also note that the rectangles at the equator are not square. This is because the X direction was mapped 360° around the entire sphere, whereas the Y direction was only mapped 180° from pole to pole.

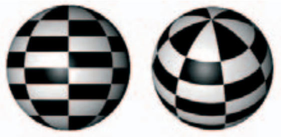
Challenge: Make the rectangles at the equator look more like squares. Once you have succeeded, return to the 4 X 4 tile (frequency) setting.



Let's prepare to see how other modes of projection affect how the texture is placed (mapped) onto the object.

Switch to the side view; be sure to set the display to *Gouraud Shading* so you can see the real time texturing in the editor. Move in on the view to make the sphere larger.

From this view the height of the rectangles looks smaller near the poles. This, of course, is because you are viewing the sphere from an angle.

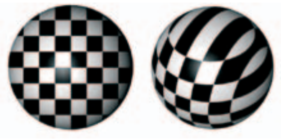


Change the projection type to *Cylindrical*.

Now the heights are all equal from top to bottom.

Switch back to the perspective view.

The checkerboard was kept uniform around the sides at the expense of more distortion at the ends.



Change the projection type to Flat.

This probably looks a bit bizarre from the perspective view.

Rotate the view to look at the sphere more from the front (down the Z-axis).

That should look better. In fact, it should look like a checkerboard! Flat is the default and only projection type that makes sense for such flat surfaces as floors and backgrounds.

Undo your view change. Change the projection type to Frontal. Rotate the view.

This mode works like a projector fixed to the camera. There are special uses for this such as some types of compositing.

Change the projection type to Shrink Wrapping. Rotate the view.

Shrink wrapping puckers all the distortion at one end. This would work for something like an eyeball (although other methods are also used for eyes) where one side remains hidden.

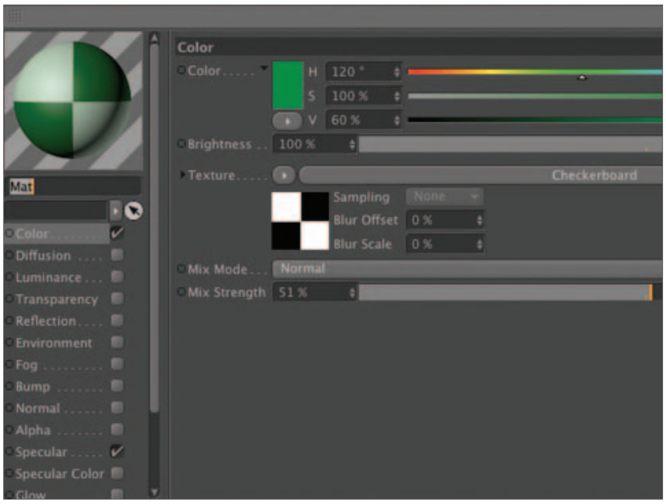
You can read more about spatial projection in the help files. UVW Mapping will be discussed in detail later. UVW Mapping is not a projection type in and of itself but rather a method of "sticking" a projection onto the surface. When the object is deformed, the texture will be deformed along with it.

Change the Projection back to Spherical before proceeding.

3. Texture mapping intro

There are two main methods of controlling how a material with textures is applied to a surface. One of these is through the *Texture* Tag, with which you can orient the texture on an object's surface. The other method is within the texture panel of the Material Editor and lets you make certain modifications to the texture itself. Lets begin with some modifications in the Material Editor.

Parameters in the Material Editor



Color

Color: H 120 °, S 100 %, V 60 %

Brightness: 100 %

Texture: Checkerboard

Sampling: None

Blur Offset: 0 %

Blur Scale: 0 %

Mix Mode: Normal

Mix Strength: 51 %

Select the Color channel of your material green in the Material Editor. Adjust the *Mix Strength* to about 50% (be sure *Mix Mode* is set to *Normal*).

The checkerboard texture has been blended with the green color you created in the Color panel. A value of 100% means the checkerboard texture completely obscures the color. A value of 0% will completely remove the checkerboard's influence on the color.

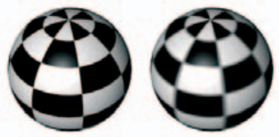
Try the other blending modes and a few other strength percentages to get a feel for their effects. When done, return the strength back to 100% and the mode back to *Normal*.

When texture maps and many 2D shaders are enlarged, e.g. when zoomed in on a surface, the graininess of the texture becomes apparent. CINEMA 4D has procedures for filling in pixels when a texture map has been enlarged beyond its normal resolution. CINEMA 4D estimates the color values of additional pixels it must add to stretch the image in a process called interpolation. Although there are several modes of interpolation available, the default MIP method offers good results with reasonable rendering times. The interpolation mode is referred to in the Material Editor as Sampling .

An example of a rare opportunity to use another interpolation type is when placing an image on a TV screen for a close-up where you want the graininess (pixels) of the video image to show. In this case you may want to use the None mode.

Below the text box for *Sampling* are two boxes labeled *O* and *S*. These are for setting the **O**ffset and **S**trength of how the image is blurred due to the sampling. You can use this to soften texture maps and 2D shaders when the unmodified texture seems too harsh.

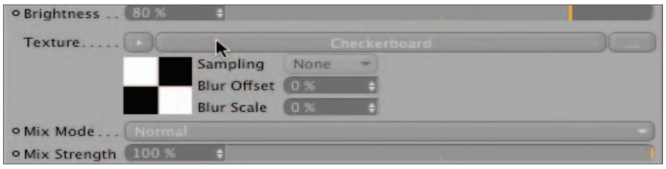
Try changing the Blur Offset to 20 and render. Return to zero when done.



Checkerboard is a shader. It is created from an algorithm that calculates how the surface will look based on parameters you can adjust.

Click on either the shader sample or the shader's name in the box right of the Texture arrow.

This panel lets you edit the shader or texture itself. With regard to texture maps (bitmap), such as a photo or an image created in Photoshop, you will not be able to do any editing. With regard to movies, you can make certain adjustments such as how the movie is looped if you want it to repeat during an animation. In the case of shaders, this is where you access their adjustable parameters.



○ Brightness . . 80 %

Texture. . . . . Checkerboard

Sampling None

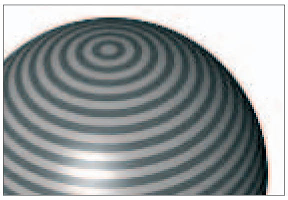
Blur Offset 0 %

Blur Scale 0 %

○ Mix Mode . . . Normal

○ Mix Strength 100 %

Change the U Frequency to 0 and the V Frequency to 8 (remember, some other mouse or keyboard action will be necessary to view the effects of entering the last value).



You probably didn't anticipate the ability of the checkerboard shader to be used this way. As with modeling, good texturing frequently involves creative use of texture maps and shaders.

Return to the 1 X 1 checker pattern. Try changing the colors of the squares with clicks on Color 1 and Color 2.

You may have noticed that there are two methods of changing the size of the pattern on your sphere. If you use *Tiling* via the *Texture* Tag, the changes will only apply to the object possessing that tag. If you change the *U* and *V Frequency* of the shader in the Material Editor, the pattern will change for every object to which the material is applied.

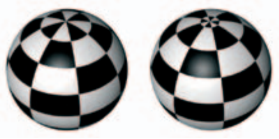
Texture Tag Parameters

Let's survey some positioning parameters accessed through the Texture Tag.

Close the Material Editor and select the Texture Tool (referred to as the Use Texture Axis Mode in the status bar). Make sure the sphere is still selected.

You should see a yellow colored mesh covering a portion of the surface (you may want to rotate the view a bit). Notice how the mesh goes 1/4 of the way around the object and 1/4 way down from pole to pole. This is your original texture squeezed to 25% in both directions. This let it be tiled four times in each direction (remember setting both the X and the Y tiling to 4?). Also be aware that the axes shown are now those of the texture, which can be different from those of the object.

Be sure the Move Tool is selected; lock the Y and Z-axes. Click and drag in the editor and observe the texture's movement in real-time (do not grab near the axis handles, this will produce a different result).



You should note (in the Attribute Manager) that the Offset X value has changed, it is no longer 0. You have changed the location within the texture's coordinate system at which the mapping starts along the X-axis.

Reset the value to 0. Lock the X-axis and unlock the Y-axis. Drag down in the view, again not grabbing the axis handles.

Now you will notice the Y offset has been changed. To get a better idea of what is happening, we will turn off tiling.

Set the Offset Y to 25 and turn off tiling by clicking on the Tile check box. Rotate the view if necessary and render in the editor.

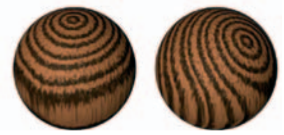
Now the origin of the texture has been shifted down (offset) and the texture only covers 25% along both X and Y directions, since it is no longer tiled. Using non-tiled textures that only cover portions of a surface is an important way to build up complex patterns using multiple materials.

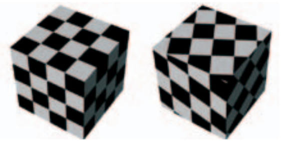
Return the Y offset to 0 and re-activate tiling. Choose the Texture Axis Tool and make sure the Coordinates Manager is open to view. Grab the X-axis handle and move it, observing the results both on the object and on the values in the Coordinates Manager.

Now that you have chosen the Texture Axis Tool, the Coordinates Manager displays the texture's position, scale and rotation, that are functions of moving its axis. This is different from using the Texture Tool that is used to reposition the location at which the texture's pattern starts in reference to the axis. These values are viewed in the Tag tab of the Attribute Manager.

Open the Coord Tab in the Attribute Manager and confirm that the values in the position (P) boxes match those in the Coordinates Manager. Undo any moves and try movement along the Y-axis. Rotate the view for a better look, if desired. Undo any changes.

With texture maps and 2D shaders (like Checkerboard) the *Offset* parameter in the Texture Tag will take care of repositioning. However, with most 3D shaders, *Offset*, *Length* and *Tiles* will not work. You must change the coordinates of the texture axes. An example of this is the Wood shader. Changing the position of the texture's axis is equivalent to moving the center of the tree from which the wood has been cut. The left sphere above has its Y-axis in the center of the sphere. The Y-axis of the right sphere is texture has been moved a bit to the right





Return all the position values to zero and select the Rotate Tool. Rotate the different axes, thereby observing the changes to the texture's orientation and to the rotation values in the Coordinates Manager.

This does not look any different from rotating the Sphere Object itself. However, if it were on a cube the difference would be noticeable. The texture would rotate but the corners of the cube would remain stationary. The ability to rotate textures is useful for shaders and for texture maps

Return all rotation values to zero and select the Scale Tool. Scale the different axes observing the changes to the texture's appearance and to the scale values in the Coordinates Manager.

As you can see, this has a different result than changing the *Length* or number of tiles. However, as discussed above, this is the method you must use to change the relative size of 3D shaders. Of course, most 3D shaders allow control of scale in their own editors, but the ability to adjust in the Texture Tag has advantages. An example might be a wood grain you wanted to use on different objects with a slightly different grain size. You have now had a brief survey of how an object is textured and the parameters controlling the texture's placement. It will take much practice to incorporate all this information usefully. We will do more specific texturing assignments later to help gain familiarity with these techniques.

4. Other material channels

Finally, back to finding out about those other channels and making our material more interesting. Let's begin with the *Diffusion* channel. Most objects look unnatural with a solid, pure color. In reality they will usually have slight variations in the color tone or may be smudged or dirty. This is what diffusion is for.

Diffusion

Reset the tiling to 1 in both the X and Y directions. Select the Model mode so that the texture mesh does not clutter the screen. Open the material green and, in the Color channel, disable the checkerboard pattern by moving the strength slider down to zero, returning to the green color.

You could also select the text "Checkerboard" and hit *Delete*. This method allows us to easily reactivate the shader again later and will preserve any adjustments made to its parameters.

Activate the Diffusion channel. From the Texture submenu, select the Surfaces / Marble shader. Render your object.

Not too subtle! This is more than just dirt - it needs a lot of toning down.

Set the *Mix* strength to about 8 and render.

The surface now has a subtle discoloration. So far, diffusion does not seem to be anything special. You could have also done this with an image in the color channel.

Back in the color channel, set the strength of the checkerboard texture to 60%. Render.

The diffusion channel allows us to dirty up the surface with or without a texture in the color channel.

Disable diffusion and render the object for comparison.

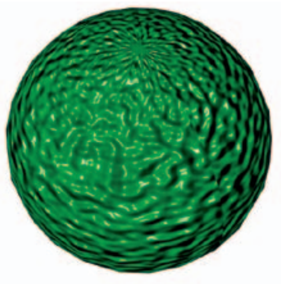
Floors, walls and many other objects benefit enormously from subtle use of diffusion.

Bump

Since very few surfaces are perfectly smooth, the bump channel is one of the most important channels.

Return the Checkerboard's influence to zero (Mix Strength = 0), disable diffusion and activate the Bump channel. Select the Marble shader to use for the Bump channel. Set the Strength to 30 and render.

You may have noticed that, unlike the checkerboard shader, the marble shader does not appear distorted at the poles of the sphere. This is because the checkerboard is a 2D shader that had to be wrapped (projected) around the object. The marble shader, on the other hand is 3D and acts like a solid block of marble from which the object is cut.

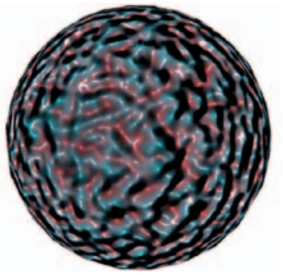


At the right is a 2D noise shader. Notice the pinching at the top where the texture is forced to come together to wrap the sphere completely. If you look carefully you will also see a seem running from pole to pole where the texture starts and ends. The lower sphere has had a 3D noise shader applied. Notice the ends are as non-distorted as the rest of the surface and no seems.

Let's do two things before moving on to other channels...

Replace the marble shader with Noise from the Texture menu. Click on the sample or name to edit. Click on the arrowhead after Noise.

Choices upon choices. *Noise* is probably the most commonly used shader, especially in the Bump channel.



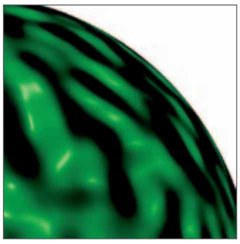
Either click on the Bump channel's name or the left open arrow at the top of the Material Editor to return to the main Bump parameters. Set the Strength to 100. Render.

Many parameters in CINEMA 4D, including bump Strength can be set to a value greater than 100 by manually entering values into the text box. Let's say we want colored bumps.

Click on the Texture menu and choose Copy Channel. Go back into the Color channel with a click on its name. Under the Texture menu, choose Paste Channel.

This will copy all adjustments you have made to parameters along with the shader.

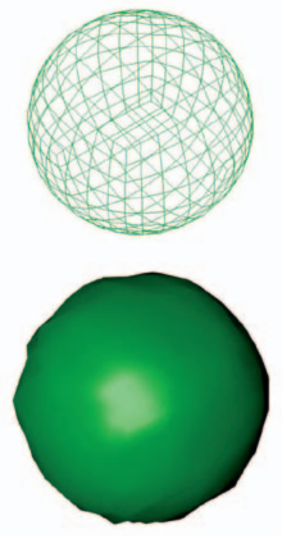
Set the Mix Strength back to 100 and click on the shader sample to edit. Change both colors to contrasting values and render.



The color and bump channels fit perfectly! You can copy and paste between any channels using textures. Before we leave bump, you should note that the bump effect only appears on the surface, not along the edges (image at left). In this respect it behaves very much like the *Phong* Tag. If you want the effect to actually affect the 3D shape, and therefore the silhouette of the object, you must use the

Displacement channel.

Displacement



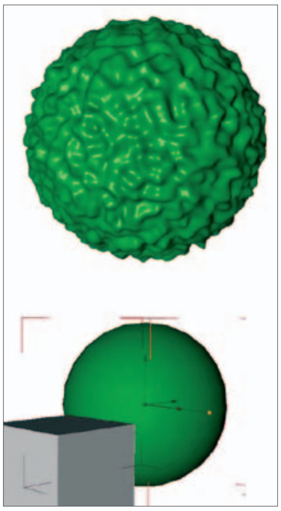
Delete the texture from the *Color* channel with *Texture / Clear*. Copy the texture from the *Bump* channel into the *Displacement* channel. Disable the *Bump* channel.

The sphere still appears smooth. Why? The displacement function requires polygons on which to operate.

Make a copy of your sphere and hide the original in both the editor and renderer. In the Attribute Manager, change Type to Hexahedron. This will result in rectangles for better results. Make the visible copy editable. Render.

Now that your object is polygonal, it can be "displaced". The effect is relatively crude because of the low polygon count.

Set the display mode to Gouraud Shading (lines) and note the number and size of the polygons in the mesh. Subdivide using Mesh / Commands / Subdivide, changing the Subdivision value to 3. Now note the size and number of polygons. Render.



This is probably more what you had in mind. This can, on occasion, save much modeling time but should be used sparingly due to the necessity of the high polygon counts. Try rotating the view to see the price in speed you have paid for all those polygons. The remaining channels need some additional objects to better see their effects.

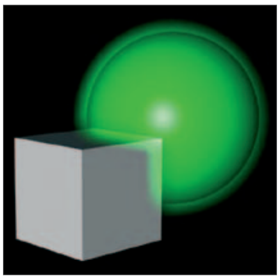
Create a Cube Primitive 100 units on a side.

New shortcuts: the 4 key with the mouse will translate the object just as the 1-key and mouse translate the view. Use the Cmd-key with the 4 key to move the object toward or away from you. The 5 key with the mouse will scale the object, the 6 key will rotate the object (4, 5, 6: not too much harder than 1, 2, 3). Position the cube between the sphere and your view so it obscures about half of the sphere (right figure).

Glow

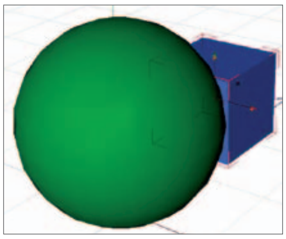
The glow effect is good for neon lights and many other effects although it has its limitations.

Delete the polygon sphere and make the primitive sphere visible. Disable the Displacement channel and activate the Glow channel. Leave settings as they are and render. Disable Glow when done experimenting.



You will notice the glow effect is added at the end of the render (a "post effect"). This happens with several other types of effects like motion blur and some lighting effects. What this means for now is that if you prematurely stop the render, you will not see the effect. The other noticeable characteristic of the glow is that it bleeds onto the edge of objects in front of the sphere, in this case, the cube. This limits the usefulness of the glow function. Plan ahead if you wish to use glow. More can be found on glow in the help files.

Transparency



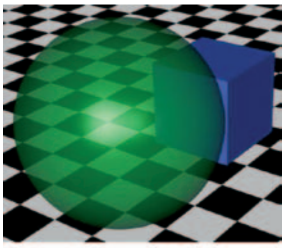
To see the effects of transparency, we should have objects placed behind the sphere. First we will create a floor object.

From the main menu, select Create / Environment / Floor or select the Floor icon from the pop-up menu attached to the horizontal perspective grid object icon in the toolbar.

The floor object is, by default, created at a height of Y = 0 which is a good place for the floor. However, this means our sphere, with its center also at 0, 0, 0, is protruding through the floor.

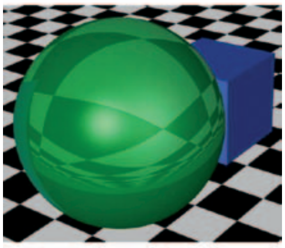
Be sure the Model mode is selected and select the sphere. Note that its size in the Y direction is 200. Since 100 of that is below the floor, enter 100 into the Y Position text box to lift it up.

Now we will position our cube behind the sphere and on the floor.



Place the cube on the floor by setting its Y position to 50. Lock the Y-axis and use the Move Tool to move the cube behind the sphere. Toggle to the top view to be sure the cube is not inside the sphere, then switch back to the perspective view. Move your cube so part of it peeks out from behind the sphere (right figure). Make a new material named "cube", with the color set to HSV of 240, 100, 100, and apply it to the cube.

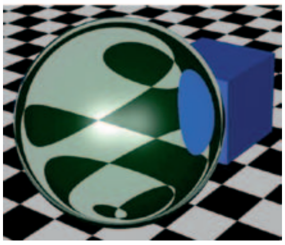
We need to texture the floor to be able to see the results of transparency.



Create a new material with the Checkerboard shader. Change the U and V Frequency both to 2. Name the material "floor" and apply it to the floor object. Notice the Flat projection type is automatically selected for the Texture Tag of a floor object. Now for the green material - activate its Transparency channel.

If the Material Editor was already open, all you need do is click on the green material's icon in the Material Manager to make it the editable material.

Set transparency to 50 by adjusting the Brightness slider. Render.



The floor and cube should become visible through the sphere (figure at right). A texture can also be used in this channel to make only certain areas transparent. You could, for example, use this to make windows made up of several smaller panes, separated by mullions. What if this were a glass ball that bends light like a lens? The degree to which it bends light is measured by a parameter called Index of Refraction (n), which is typically about 1.5 for glass.

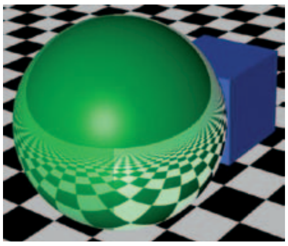
In the transparency panel, change the Refraction value to 1.5.

The "glass" sphere acts like a strong lens that magnifies to such a degree that only one square in the checkerboard floor takes up almost half the view through the sphere. The rays of light from the part of the cube behind the sphere are bent (refracted) so much that they do not make it to your eye. Another common material that refracts light is water (n = 1.33). Values for other materials can be found in the Help files or in a physics text. We will need to use a weaker index of refraction to actually see the cube through the sphere.

Change the transparency value (brightness) to 90, set *n* to 1.15 and render.

Reflection

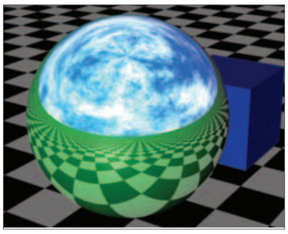
Whereas *Specular* reflects lights off the surface, *Reflection* shows images of other objects as they would appear in a shiny metallic or glass-like surface.



Disable the Transparency channel and activate the Reflection channel. Render. Change the Brightness value to 50 and render for comparison.

Here again, the value is defined by the *Brightness* slider. You will note the reflection of the floor around the bottom part of the sphere. However, the top looks unaffected. This is because there is nothing to reflect from the top of the scene. One quick way to add something to reflect, to emphasize that the object is uniformly reflective, is by using the *Environment* channel.

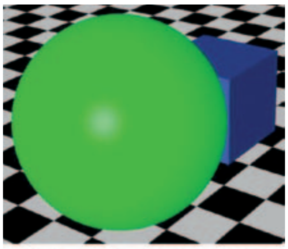
Environment



Here you can add a texture that will add some variation to the surface. This will help convey the sense of other elements in the scene, lending to the reflection without actually having to create it. This is especially useful for metals. Be careful to use it subtly, though.

Return the reflection Brightness back to 100. Activate the Environment channel and select the Cloud shader (Surfaces / Cloud). Render.

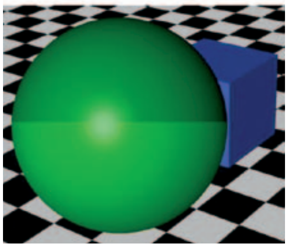
Luminance



Luminance will cause objects to appear as if they are emitting their own light, independently of the lights in a scene. This will only affect the object itself and will not illuminate other objects as lights will. Subtle use of luminance will reduce contrast by slightly illuminating the dark, shadowy areas of an object.

Disable the Reflection and Environment channels and render. Activate the luminance channel and change the color to H = 120, S = 100, V = 60 (the same as for the Color channel). Render.

This is a pretty strong use of luminance.



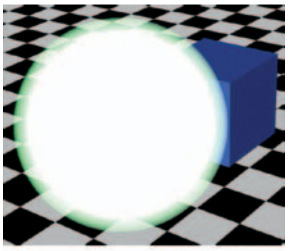
Change the Brightness value to 25 and render. Disable luminance and render for comparison (figure at the right shows top half without luminance, bottom half with 25%).

You will notice reduced shadow and contrast on the object when luminance is active. Luminance is also excellent for such objects as lamp shades. By using a color other than that in the *Color* channel, you can obtain special effects.

Reactivate luminance and change the color to H = 0, S = 100, V = 100. Change the *Brightness* to 35 and render.

In the shadow areas, where the green surface is not illuminated by the light, the red luminance dominates. At the front, where the light shines, the surface color is a combination of the green color plus the red luminance.

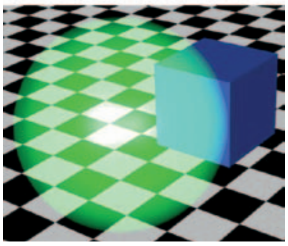
Fog



This lets you generate local areas of fog confined to the volume of the object to which it is applied (there is a global fog function that will make your whole scene foggy). Here is just a quick glimpse.

Disable Luminance, activate Fog and render.

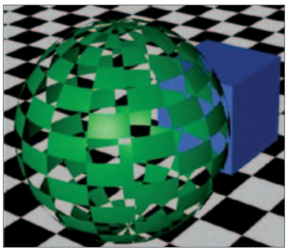
The Distance value indicates the distance at which a light ray is completely “weakened” by the fog. The lower the value, the thicker the fog (upper right).



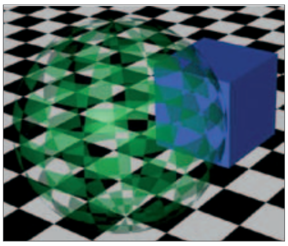
Try a Distance value of 250, render to view the result.

In addition to the *Fog* channel and the environmental (global) fog, there is a fog shader which lets you create a gradient as is the case with fog that is thicker near the ground and thins out as it rises. Investigate more in the help files when you feel the need to use fog.

Alpha



This last channel is different from the rest and is indispensable in creating complex textured surfaces using multiple materials. You can probably figure out one way to apply two materials to the same surface. The first could cover the entire surface, the second could have both X and Y lengths set to less than 100 with tiling turned off. This would be good for such hard-edged breaks in the surface texture such as a label on a can, but what if you wanted an irregular or softly graded transition to a new texture? Alpha channels contain



information as to where a texture should be made transparent so the underlying texture shows through at certain locations only (or so the object completely disappears from those areas).

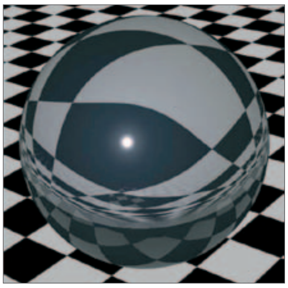
Disable the **Fog** channel and activate the **Alpha** channel. Select the **Checkerboard** shader and set both its **U** and **V Frequency** to 2. Render.

Areas of the image or shader used for the alpha that are black have been completely hidden. Since there is no other texture underneath green on your sphere, these areas are no just made transparent, they simply disappear. If your were to reactivate transparency, there would be no refraction in those areas. Furthermore, alpha is not an all or nothing effect.

Disable the Image Alpha function. Edit the checkerboard texture, changing the Value (V) of Color 1 to 50. Render.

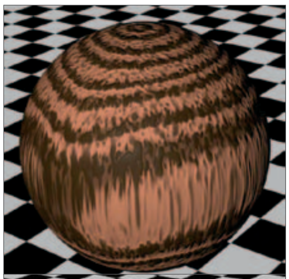
You have just made the previously white, opaque areas gray (50% black). They have now gained a degree of transparency. We will do much more with alpha channels. Alpha channels are important enough and somewhat complicated enough that you should read and study what the Help files have to say.

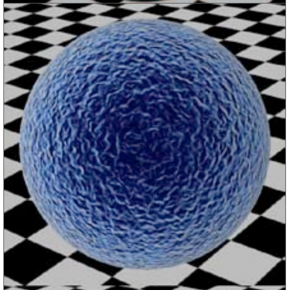
5. 3D Shaders

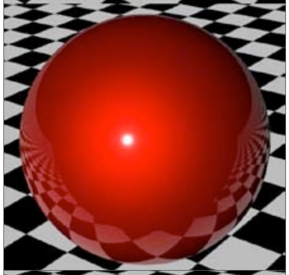


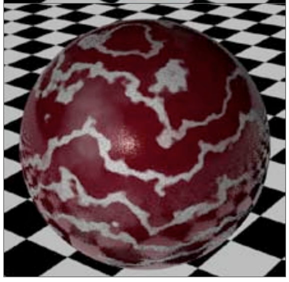
We will conclude this tutorial with just a glimpse at some 3D, or volumetric, shaders. There are two principal types: channel shaders (such as Noise and Checkerboard) that are used in channels in the Material Editor and volume shaders that are accessed through the Create menu in the Material Manager.

The names of the shaders are a bit strange. They were created by some strange and talented guys. Following is a brief description of their principal function, but some of them are very versatile and can produce quite a range of surfaces.









- Banji: glass
- Banzi: wood
- Cheen: translucent, backlit surfaces
- Daniel: reflective surfaces
- Mabel: marble
- Nukei: What to say about Nukei?

To get an idea of the range of available surfaces we will sample the default settings for each shader.

Create a new material from the Material Manager using Create / Shader / Banji. Name the material "Banji" and apply it to the sphere by dragging its icon from the Material Manager onto the existing Texture Tag for your sphere. Render.

Banji makes good glass. You will notice transparency with refraction as well as reflection. You can come pretty close to this by adjusting the channels in the Material Manager but Banji offers subtleties such as increased edge control and more. Dragging the new material onto an existing tag as instructed above is fast. It will also result in the new texturing inheriting any changes you previously made in the Attribute Manager for the Texture Tag.

Select the Sphere's Texture Tag.

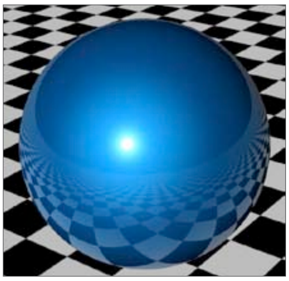
You should notice (in the Attribute Manager) that the tiling is still set to 4 for both X and Y directions, as you set previously in this tutorial.

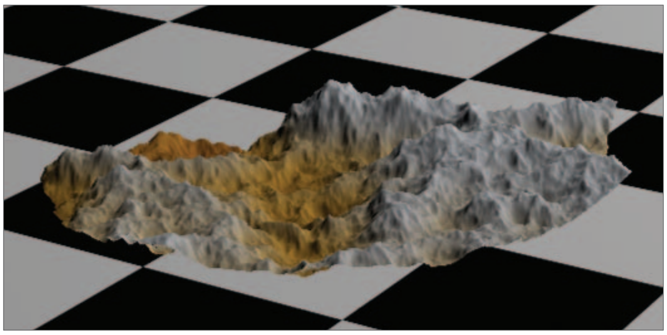
Open Banji's editor with a double-click on its icon.

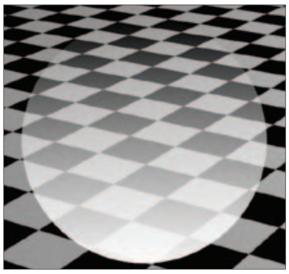
Most volume shaders have their own, unique editors. Finish the survey of the other volume shaders. If you have time, open some of their editors and experiment with the parameters.

Create five other materials, one for each of the remaining volume shaders. Give each material the name of the shader as you did for Banji. Apply each in turn to the sphere and render.

Time permitting, experiment with the parameters of some of the volume materials. We will use Banzi in another tutorial to give your table a wood look. This tutorial was only an introduction to materials and texturing. The look of an object is heavily dependent on quality texturing. Complex surfaces generally require more than one texture combined or layered in various ways. We will visit and practice some of these techniques later on in various tutorials.







Texturing and Label Tutorial

In this tutorial you will create a plastic-like material for the surface of your previously created bottle. You will then place a label on the bottle.

The Bottle Label

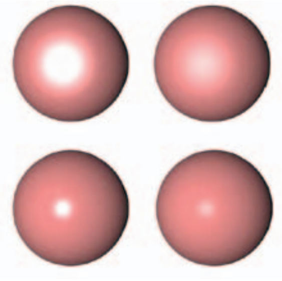
Open the file "Plastic Bottle 1" which you previously created. Immediately do a *Save As*, saving the new copy as "Plastic Bottle 2" also within the "Plastic Bottle" folder.

This is done to preserve the original copy; any subsequent saves will now be done to this new file leaving the other in its original state as a back-up. With serious, complex projects you may find yourself up to version 50 or more before You are ready for the final render.

Create a new material named "bot" and open the Material Editor. Change the Color HSV values to 180, 5, 95 with Brightness set to 100.

Brightness has a similar function to Value except that it can be animated separately from the Color parameter, which is more involved. The values entered will produce a slightly off-white color similar to that found in many plastic milk bottles. Feel free to experiment. The principal consideration is to use a low *Saturation* value if you are after some shade of off-white.

Set the Specular values to Width = 70, height = 20.



This type of plastic isn't too shiny, hence the *Height* value (intensity) of only 20. However, plastic of this nature has microscopic pits over its entire surface that reflect some portion of the light to the camera, hence the increased *Width* value. Again, feel free to experiment. In the figure at the right, the top two spheres have a large *Width* value, the two spheres on the left have a large *Height* value

Apply the material to the bottle, leave the default projection as is. Render in the editor.

Since there are yet no texture maps in the material, the projection type is immaterial. The color, for most applications, is a little too uniform. Let's add some diffusion.

Activate the Diffusion channel and select the Noise shader from the Texture menu (arrowhead).

This is probably the most useful shader for use in the Diffusion and Bump channels. Since it is volumetric (3D), it is applied without seams or pinching (since it has its own mathematical formula, it is applied to the surface independently of the projection type selected in the Texture Tag). We now need to edit this shader which, like all shaders, has its own editor.

Click on the shader sample or name to edit the shader. Click on the name of the Noise parameter and select "Cranal" from the pop-up list. Change the Global Scale to 200.

We could have also adjusted the scale using the Texture Tag. However, since we will introduce another shader in the Bump channel, we chose to do the adjustment here so the two channels can be set independently of one another.

Return to the main Diffusion panel, set the Mix Strength value to 50 and render.

Far too strong, but you get a sense of the look. Let's make it more subtle.

Change the value to about 10. Render. Turn diffusion off and render for comparison.

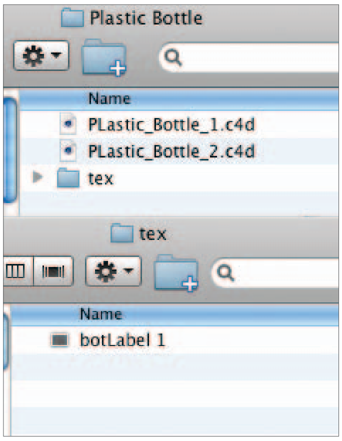
You should note that the look is subtle, yet it makes a difference. Leave diffusion off for now so you can better see the effects of the bump we are about to create.

Activate the Bump channel and again select Noise. Leave the Noise type to the default Noise but change the Global Scale to about 15. Back in the main panel, set the Strength slider to a value between 1 and 3. Render.

You should now see the typical surface characteristics of polyethylene plastics. Reducing the scale made the pebble size quite small, which is characteristic of this type of plastic (time permitting, try different values).

Reactivate the *Diffusion* channel and render.

Your basic bottle texture is complete for now. Next we will place a label on the front of the bottle that wraps partly around each side. This label was created using a drawing program and saved as a TIFF file. One of the elements used to create the label image was a CINEMA 4D rendering of the bottle itself. The TIFF file will be used as a texture map in the channel of a new material that will make up our label.



Plastic Bottle

Name
PLastic_Bottle_1.c4d
PLastic_Bottle_2.c4d
tex

tex

Name
botLabel 1

Before we create the label material, we need to discuss how CINEMA 4D keeps track of images it uses for textures. There is a set of textures that were supplied with CINEMA 4D, that are kept in a folder labeled "tex" within the MAXON folder in the CINEMA 4D directory. CINEMA 4D always knows to look there for textures. This is where you would place stock textures which might be used again and again in many different files. These might include concrete, cork, brick and many more. However, many files will have specialized textures (bitmaps) that you have created only for use in that file or a few others. The standard practice is to keep these textures within a folder contained in the same folder as the file itself. This folder must also be named "tex" and must be at the same level in the folder as the file which uses the texture (figure at right). This is a good idea, even if it means sometimes having duplicate copies of the same image in two or more "tex" folders, because it makes the file more portable. If you want to transfer it to another computer, you can transfer the entire folder, including the textures. If these textures are not present in a "tex" folder within the file folder, or in the main "tex" folder, CINEMA 4D will not be able to render them and you will get a "texture error".

Return to the desktop. Within the "Plastic Bottle" folder, create a new folder named "tex". From the "tex" folder within your "Student" folder, drag a copy of the texture "botLabel_1.tif" into the new "tex" folder.

In most cases, textures this specialized should not be copied since this means wasting extra disc space.

Back in CINEMA 4D, create a new material named "label". In the Color channel, click on Texture / Load Image and navigate through folders until you find the folder named "tex" within the "Plastic Bottle" folder. Select the image titled "botLabel_1.tif". Change the Specular values to Width = 40, height = 60. Close the Material Editor.

These values will make the label appear smoother and shinier. Now we are ready to place our label on the bottle.

Apply the label to the bottle by dragging its material onto the bottle object in the Object Manager. Change the projection type to Cylindrical.

This is a bit tricky here. Many labels can be applied using a flat projection but this one wraps around the two sides as well as the front which would cause a flat projection to warp. You will also notice that the second texture (label) was placed to the right of the base texture (bot) in the Object Manager. This means it will be placed over bot which is exactly what we want.

In the Attribute Manager, with label's Texture Tag still selected, disable the *Tile* check box. Rotate the view so you can see the right side of the bottle.

We do not want our label repeating itself to cover the entire surface.

Looking at the right side of the bottle, you will see that the label wraps all the way around the bottle once (in the X direction). We only want it to wrap about 60% to 70% around the bottle.

Select the Texture Axis mode, choose the Scale Tool and lock the Y and Z-axes. Drag in the editor and notice the label shrink (or expand) along the X direction (around the bottle). Note also in the Attribute Manager how the values for X Length and X Tiles have changed. Set the value of the X Length to 70.

This should make your label about the right width. If not, you can adjust it later. Let’s bring it around to the front.

In the Attribute Manager, drag the arrowheads associated with the *Offset X* value to move the label around the bottle until it is centered on the front. You may need to rotate the view or, better yet, switch to the front view.

The label seems to be placed a bit high.

Lower the label (not yet lowered in image to right).

The Bottle Label
...and you can do it too!

We could stop now since our bottle is looking pretty good. However, let’s make one additional tweak just for practice. Close examination will probably reveal that the writing meant to be on the sides of the label carries around the corner a bit to the front. If we use the *Texture Tool* to scale the label larger, it will become too large and will wrap around into the handle region in an unnatural way. We can accomplish our tweak using a couple of adjustments to the texture’s axis.

Select the Texture Axis Tool and the Move Tool. Move the texture back just a bit along its Z-axis only. Check the Coordinates Manager (since this move involved the texture axis) to see which value has changed. The Z Position value should be about 9.

This is a little better but has also dragged the texture back uncomfortably close to the handle. Let’s squeeze it back.

With the Texture Axis Tool still selected, choose the Scale Tool. Slide the handle of the Z-axis toward the center of the texture just a bit. Check the Attribute Manager again to see which value has been affected (the Z Size value should be about 95).

The Bottle Label
...and you can do it too!

These tweaks should be used sparingly since they result in some distortion. You will notice the front of the label is now a little more weighty. You could increase the height of the label a bit to compensate, although the text on the sides would stretch vertically as well. If you really need to get that meticulous, then you should have planned the construction of your label’s image more carefully...

It is possible to place several textures on a single surface. The underlying textures show through where desired because the uppermost textures are either:

- Reduced in size but not tiled,
- Restricted to certain selected areas, or
- Have alpha channels making portions of them transparent to the textures underneath.

If you are quite a bit ahead of the group, and you have CINEMA 4D’s Advanced Render module, try adding a little *Subsurface Scattering* in the *Luminance* channel of the bottle. Refer to the help files for details.

Table Shader Tutorial

In this tutorial you will texture the wooden table you made earlier, using one of the volume shaders.

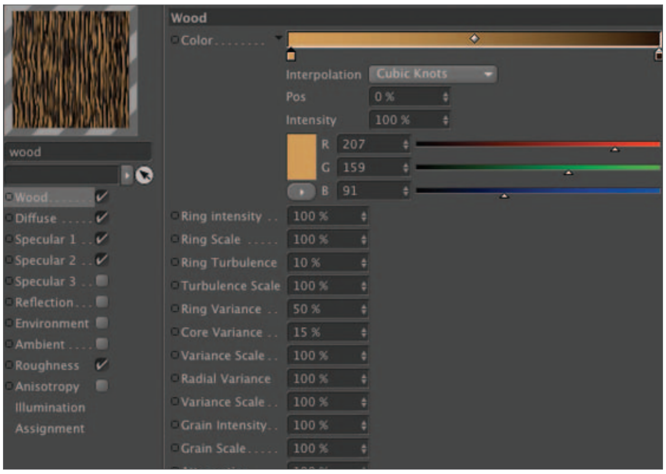
Here we will texture the table you created in the modeling tutorial with a wood grain material. We will create this material using a 3D volume shader with its own unique editor.

Open the file "Wood Table 1.c4d" and Save As "Wood Table 2". Hide all objects except for the tabletop.

The best way to do this is to hide the entire table and display only the top (green).

From the menu of the Material Manager, choose Create / Shader / Banzi. Open the Material Editor and change the name of the new material to wood1.

Notice the different channels. At the top of the panel in the *Wood* channel, after the parameter name *Color*, is a color bar called a "gradient". Gradients are used to control many parameters in CINEMA 4D. This one specifies the wood's colors from the summer growth (base color to the left) to the growth rings (at the right). A click on the tiny arrowhead brings up the color editor. The tiny square color samples under the gradient are called "knots". A click on the knot makes it active, allowing you to adjust its color. You can also slide these knots to different positions along the gradient. A click in an empty position under a gradient will add another knot. Drag a knot downward to remove it.



Wood

Color

Interpolation: Cubic Knots
Pos: 0 %
Intensity: 100 %
R 207
G 159
B 91

wood

Wood ☑
Diffuse ☑
Specular 1 ☑
Specular 2 ☑
Specular 3 □
Reflection □
Environment □
Ambient □
Roughness ☑
Anisotropy □
Illumination
Assignment

Ring intensity: 100 %
Ring Scale: 100 %
Ring Turbulence: 10 %
Turbulence Scale: 100 %
Ring Variance: 50 %
Core Variance: 15 %
Variance Scale: 100 %
Radial Variance: 100 %
Variance Scale: 100 %
Grain Intensity: 100 %
Grain Scale: 100 %

To better see changes as we change parameters, let's enlarge the preview so we do not have to render so often to see the results.

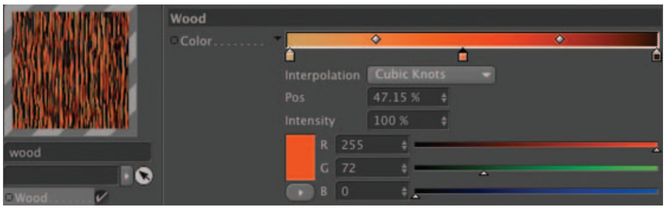
Right-click on the preview icon and change the size to Large and then change the type to Plane.

The Banzi default is a dark walnut blend. Let's adjust it to see how these knots work. First we will change the lighter color, which corresponds to the summer growth rings of the tree.

Open the color editor with a click on the arrowhead to the right of color, then click on the knot under the far left end of the gradient. Lighten the color to a pale straw color more characteristic of pine (HSV 40, 40, 90). Close the color picker and view the results in the preview box.

In addition to the knots along the bottom, the "bias handles" (diamonds) within the gradient bar slide to change how the colors are distributed.

Move the bias handles left and right to see how the preview changes (you must release the mouse to see each change). Also try sliding the knots left and right to see the effects.



Wood

Color

Interpolation: Cubic Knots
Pos: 47.15 %
Intensity: 100 %
R 255
G 72
B 0

wood

Wood ☑

Although these two colors are enough for most wood textures, you are not limited to just a two-color gradient. Simply click just below the gradient bar to add a point, drag it down to delete it.

Return the knots and bias handles to their former positions. Apply the material to the tabletop.

When using volume shaders, the projection type is ignored. The shader calculates the texture using its own internal formulae. The *Projection*, *Offset*, *Length* and *Tiles* input fields of the *Texture* Tag in the Attribute Manager are also ignored.

Also, do not bother with the *Texture Tool*. You should use only the *Texture Axis Tool* to adjust the *Position*, *Scale* and *Rotation* of the shader either using the Coordinates Manager, the Attribute Manager or the interactive handles.

Render the top in the editor (be sure antialiasing is *None* **in the render settings).**

First, you will notice the shader is not previewed on the object in the editor. You must render to see the results. Second, you see the grain of the wood running vertically at the top, a rather unlikely cut of wood for a tabletop. Let's rotate the grain to a more natural orientation.

Choose the Texture Axis mode. Make sure the Texture Tag is selected and rotate the texture to a pitch (P) of 90° so the texture's Y-axis is pointing length-wise along the surface of the top. Render.

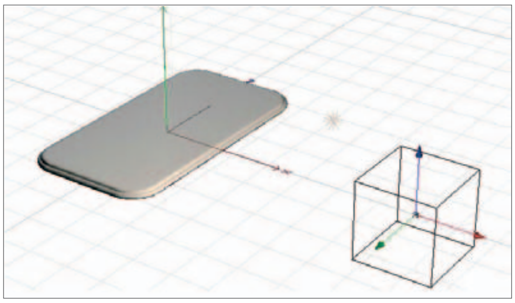
You might have noticed that the grain appears a bit coarse.

In the shader editor, change the Ring Scale to 30 and render a region.

You have to be careful with the use of shader parameter settings. Values that look good up close may show artifacts resembling moiré patterns when viewed at a distance. It is always best to judge final values at the distance the object is to be viewed.

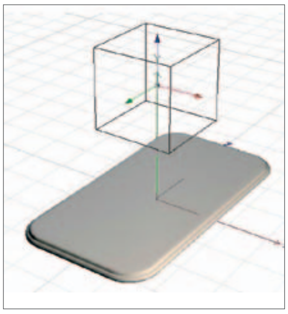
The growth rings in a tree represent seasonal differences in growth rates. The lighter, softer and wider rings are the spring or summer growth when conditions are favorable. The thinner, harder and generally darker rings are from the slow growth of late season. Let's reduce the wood's grain so we can better see the pattern of rings.

Change the Grain Intensity to 10 and render the entire top.



Pay close attention to which part of the tree the wood for the top comes from. That is the core of the tree running down through the center of our table. This is not a very good cut of wood. It is subject to warping and cracking. Let's go for some quarter sawn wood where the rings pretty much run vertically in the boards when seen from the end, as they currently appear along the long edges of our tabletop.

Back away from the table in the view so the table occupies about 20% of the view and sits at the left side. Using the Texture Axis Tool with the Move Tool, move the texture to the right side along its X-axis until it is about 700 units away. Move the view back in on the table again and render the entire top.



The opposite is slab sawed wood. Slab sawed wood is less stable but does reveal more interesting patterns.

Set the X position of the texture axis back to 0 and move the texture vertically (along the Y-axis) until the Y value is about 300. Render the entire top.

By now you should see the relationship between the texture axes and the growth rings: The Y-axis of the texture is pointing straight up the center of the tree. The growth rings wrap around the Y-axis as concentric cylinders. Moving the texture around is like moving the tree around until the grain we want is positioned within our object. Let's add some grain back to our wood.

Change the Grain Intensity back to 50.

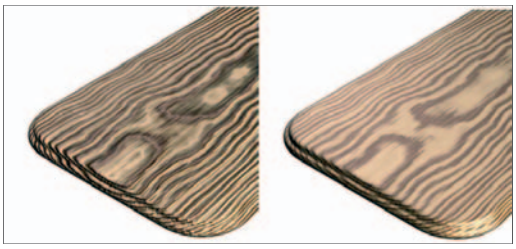
A varnish or clear lacquer on our wood would add a shine.

Activate the **Specular 3** channel, change the color to 30, 20, 100 and set the **Intensity** to 100. Render.

Other considerations affecting the shine on the table are the direction from which lights will reflect off of your table and how smooth the surface is. You may want to adjust the specular values or you can apply values greater than 100 for a really shiny surface. Notice how the grain on the table is raised? This is typically a property of weathered or rustic wood. A quality table would be more smoothly sanded.

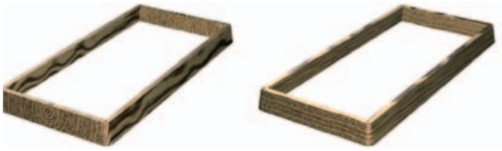
In the **Roughness** channel, change the **Amplitude** to about "15" and render (have you been saving?)

Now it is very smooth.

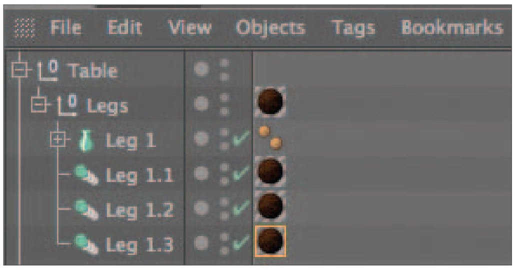


There are other channels and parameters you can experiment with. To truly develop the impression of a finish, you might add reflection as well as adjusting the specular values. Anisotropy is the fine "scratchiness" of a surface, especially useful in burnished metal surfaces but also present in some wood finishes as the result of polishing techniques (imagine the effect from rubbing the surface with steel wool).

You now need to texture the rails and legs of your table. One option is to apply the same material, then reorient and, if necessary, scale it. Alternately, you could duplicate wood1, modify the copies and apply them to the rails and legs. The second method has the advantage of allowing you to individually adjust such parameters as grain intensity, smoothness, etc.



The rails, if real, would be made from four pieces of lumber. The figure at the left below looks as if it were carved out of a single piece of lumber, not very realistic looking. The figure at the right appears much more realistic, although close inspection of the corners reveals that this is still "a single piece of wood. However, this will rarely be noticed. It is left to you to produce this effect by positioning and orienting the texture.



File Edit View Objects Tags Bookmarks

- Table
  - Legs
    - Leg 1
    - Leg 1.1
    - Leg 1.2
    - Leg 1.3

For the legs, you want the Y-axis of the texture oriented in line with each leg so the grain runs along the long axis of the leg. Since your legs point in different directions, this means you will have to apply the texture separately to each leg. However, if you used instances for the copies of the first leg, all instanced copies will share the same texture and orientation as the original. Here is what you should do: Make sure the first leg (original) is in the same Null Object as the instances (see the figure at right). Now apply the texture to the Null Object that you want applied to the first leg itself. Next, apply separate textures to each instance. As you drag the material onto each instance, the Y-axis of the texture will automatically line up with that of the leg. Move the texture so the leg does not look like it was cut from the center of the tree and each leg has a somewhat different look. The texture for the first leg, because it is applied to the Null Object, will need to be rotated to line up along the Y-axis. If your legs are all independent objects, simply drag the material onto each and reposition as described above.

Now you have had a taste of volumetric shaders. Their application and modification can be painstaking but they offer a powerful method of obtaining a realistic three-dimensional look.

Lamp Materials Tutorial

In this tutorial you will practice some methods you have previously learned to create and apply materials to your table lamp.

The base

Let's start with the base. There are an infinite number of possibilities. Let's look at two of these. You can choose which one you like and adjust the parameters to your liking. First, the expensive marble base.

Open the file "Table Lamp 1.c4d" and immediately Save As "Table Lamp 2". Hide all objects except for the base. Create a new volumetric shader from the Material Manager menu, Create / Shader / Mabel. Open the Material Editor.

Where is the color channel? This shader has two sets of channels - one for the veins (*surface 1*) and the other for the base color (*surface 2*).

Open (click on) the Diffuse A channel and note the color sample.

You should see white in the color sample box. This channel shows the veins (at least as the current veining parameters are set).

Change from Surface 1 to Surface 2 and open the Diffuse B channel. Unfold the Color editor and change the base color (currently red).

The color is also influenced by the three specular channels. Each of these allows control over the Intensity (brightness) and Size (width) of the specular highlights as well as the specular color. Generally, the first channel is the more intense, yet narrower highlight, and channel three is a more subtle, overall highlight. Experiment with these only if you have time. Let's play just a bit with the veining.

Make the preview sample larger. In the Veining channel, change the Veining Stirring parameter to 200.

Again, there are lots of parameters here and you could spend lots of time experimenting. For now, set the stirring value to something you like and let's look at just one more parameter for this shader.

Choose Surface 1 and activate the Roughness channel. Change the Amplitude to 15.

This should result in a smoother, more polished look. You may want to tweak these values after seeing the texture on your lamp base.

Try a different color for the veining (surface1) if you wish. Name the material base and apply it to the base. Render in the editor.

The material may look a bit to large for the base. You could try to adjust this in the shader's editor but we will adjust it using the Texture Tag. Remember, Tiling, Length and Offset do not work with shaders.

With the Texture Tag selected, open the Coordinates tab in the Attribute Manager. Change the scale values all to 20. Render.

You can do this in the Coordinates Manager but you would first have to select the Texture Axis Tool. Your base should be looking pretty good for now. You may still wish to adjust the scale once you have established the final size of the lamp in the scene.

The brass

Now we will give our metal components a brass-like texture.

Hide the base and show the socket. Create a new material with a color (HSV) of about 37, 68, 86 and a Brightness of 100 and name it brass.

This will give us more of an antique brass as opposed to a bright, common brass. Play around with the color if you have time before the next step.

In the Specular channel, change the Mode to Metal and set the Width to 80 and the Height to about 200.

This big height value with the metal mode is the principal trick to creating metal materials.

Activate Reflection and set the Brightness to 10. Apply the material to all the metal components and render to see how it looks on the socket. You may have to move the Default Light to better see the results (under Display in the view menu); click and drag in the sphere that pops up.

There are other tricks to making good metals such as using environment maps, setting the specular color to a value different from the lights and changing the illumination method in the Material Editor. Since these metal parts probably will not be seen at all, we will not tweak them any further.

The bulb

As with metals, you can spend some time trying to get glass to look right. Since the bulb will almost surely not be seen, we will spend just enough time to see the basic principles of glass. The *Banji* volume shader makes an excellent glass but we will build ours from the Material Editor to better see the nature of glass.

Create a new material named glass. Set the Color to 200, 20, 100. In the color panel, set the Brightness to 50.

A bit of color usually helps, even though glass, at first glance, often appears perfectly clear. A brightness setting of 50 may surprise you as well.

Activate Transparency and set the value (brightness) to about 80. In the Specular channel, set the Height to about 60. Activate the Reflection channel and set the value (brightness again) to around 20. Unhide the bulb, apply the material and render.

This should give us a reasonable approximation of glass for a clear bulb. Notice we didn't use any *Refraction* in the transparency channel (n= 1). This is because our bulb is hollow and would not produce the lens-like effects of a solid piece of glass.

A parchment shade

There are several possibilities for creating a lamp shade material. There are many bitmapped texture examples of textiles available (some on your computer) that would be appropriate for the lamp shade. Using cylindrical projection, these textures would not even suffer from distortion. One problem might arise if you have to tile the texture. Even though quality textures are made to be "seamless", patterns sometimes emerge if they are tiled extensively. Instead of using one of these prefab textures, let's create our shade using shaders in the channels of a material to give us somewhat of a parchment look.

Create a new material named shd1. Activate the Bump channel and select Noise for the Texture. Click on the sample to edit the noise shader. Change the Noise type to "Cranal" (from the pull-down menu right of Noise). Set the High Clip to about 80.

Notice also that the bump lines are indented instead of raised like the fibers in parchment (you may need to increase the preview size to see this).

Set the bump Strength to about -12. Copy this texture to the Color channel (using the pull-down Texture menu).

You could have recreated the shader in the color channel but this was good practice for when you have spent a lot of time getting a shader just right and need it in another channel.

Edit the colors of the shader in the Color channel. Set Color 1 to about 35, 20, 75 and Color 2 to about 30, 10, 100.

This is just to get you started. Tweak these colors later if you wish.

Activate the Diffusion channel and choose the Noise shader again. This time, in the shader's editor, leave the default type at Noise but set the Global Scale to 200.

Of course you could also scale this using the Texture Tag but then you would also be scaling the bump and color channels as well. Scaling here lets you get the proportions between the channels right.

Back in the Diffusion panel, set the Mix Strength of the shader to about 12. Apply the material to the shade fabric using Cylindrical projection. In the Coordinates tab of the Attribute Manager, change the scale values of the texture to 20, 20, 20. Show shade fabric, hide the other objects and render.

Play with the scale and color settings if you have the time. It is possible that the bump ridges are reversed from the preview. This is because the order in which your spline's points were drawn resulted in the normals of the polygons (that are calculated during rendering) being faced inward. If this happens, either change the value of the bump *Strength* to positive or reverse the Sequence of the points in the Spline:

If you need to reverse the bump, select shade Profile and choose the Point mode. From the pop-up, select Reverse Sequence.

We also need to apply the material to the rims. However, since we want the rims to be opaque, while allowing the main fabric to show some transparency, we need to duplicate the material.

Drag a copy of the Texture Tag on shade fabric to each of the two rim objects. Back in the Material Manager, Control+drag on the shd1 icon to make a new copy to the right of shd1 and rename it shd1b. Apply this new material to the rims by dragging the icon on top of the Texture Tags for the two rim objects.

So why did we copy the tags from shade fabric onto the rims only to replace the new tags with the new material? When you copy tags, you also copy the settings, such as the type of projection and scaling. When you replace the texture by dragging a new one onto an existing tag, that information is preserved.

Make the original shd1 material transparent with a strength of about 15.

A canvas shade

Here we will create another material for the lamp shade using a bitmapped texture of a canvas-like fabric. We will use a stock texture for the fabric. Before completing the material, we will ensure that we always have convenient access to this texture map.

Make the desktop (finder) active. Within the "Table Lamp" folder, Create a new folder named "tex". Within your student folder is another "tex" folder. Open this folder and create a copy (option-drag on a Mac) of "Canvas.tif" in the new tex folder you have just created.

Make sure the new tex folder is at the same level as the CINEMA 4D files so each version of your lamp will be able to access the textures. In this case, however, we are using a stock texture, of which there is a copy in the MAXON folder where CINEMA 4D can always find it. However, this is certainly not true for many texture images, nor for all computers running CINEMA 4D. Having this copy in the same location as the files makes your work much more portable and dependable (there is also a *Save Project* command in the main menu that will automatically create a new folder containing a fresh copy of your file and save all textures within a new tex folder).

Back in CINEMA 4D, create a new material named shd2. In the Color channel, use the Load Image option in the Texture menu and navigate to the canvas texture: CGA / Student / Table Lamp / Tex / Canvas.tif. Open the image to load it. Copy the image from the Color channel to the Bump channel and activate the Bump channel. Leave the bump Strength at the default of 20.

You can adjust the bump to your liking later, after applying the material to the shade.

Create a new Null Object and drag the existing Texture Tag for shade fabric onto it.

This is a way of temporarily storing the texture so you do not have to re-apply it and re-adjust all the parameters. If you are not going to add any transparency to the canvas shade, this is not necessary, as the most recently applied material (the canvas) will lie on top of any existing textures.

Apply shd2 to shade fabric and render in the editor.

Needs some tiling.

Try Tiles X = 5 and Tiles Y = 10. Render and further adjust if desired.

You will want the same ability to control the transparency of the fabric while keeping the rims opaque, as with the shd1 material.

Make a copy of shd2, rename it shd2b (for the rims) and then give the original a slight bit of transparency (brightness about 20).

If you are ahead in time, experiment with using the canvas texture in the transparency channel.

Decide which lamp shade material you prefer and apply it to the shade fabric and rims.

Keep the other materials, just in case they fit better in a scene in which you might later use the lamp.

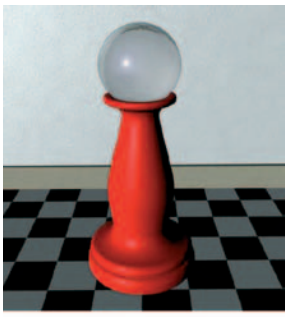
If you have made a switch for your light, give it an appropriate material such as black plastic (specular Width high, height low). Unhide the rest of your lamp, get a good view and render in the editor.

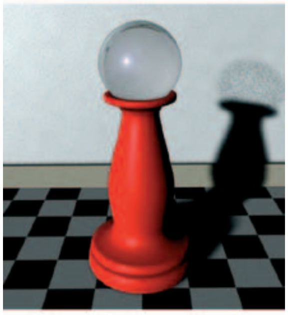
If you have time, play around with some of the textures and make your lamp unique. Try the *Danel* volume shader for the base, it will give you a bright, glossy glazed look (although, in general, you should try to avoid having a glazed look in this class). Try using the *Cloud* shader with a strength of 10 in the *Environment* channel of the brass material (edit the blue in the sky to a medium shade of green and tile the shader).

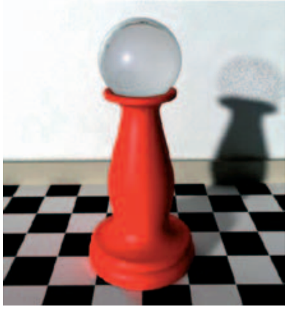
In the next unit, we will light our lamp. In doing so, you will see that you are not really finished texturing. Texturing and lighting are closely dependent on each other. For instance, placing a light inside your shade might make the shade seem too transparent. Also, due to the light from the bulb, the shade will appear to be luminous and therefore will probably benefit from use of the *Luminance* channel.

Introduction to Lighting

Overview







Lighting is almost as important in defining the look of a surface as texturing. And in some cases even more important. So far, you have been viewing scenes you have practiced modeling and texturing using the Autolight. As soon as you add an active light object to the scene, the Autolight is disabled and you are in control of all lighting parameters.

Lights are objects that can be moved and rotated, hidden, made Children of other objects, and animated as any other object. And, of course, they have special properties of their own.

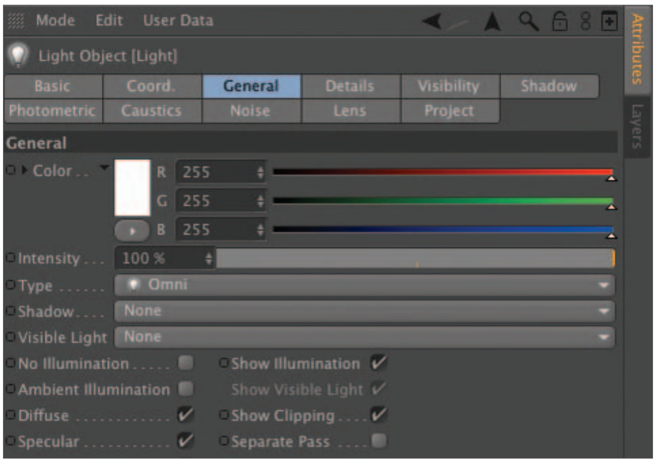
The most important of these properties is **illumination,** which is the casting of light onto another surface (top right). Illuminated objects may then cast shadows (middle right) as a consequence of the light falling on them. Lights may also be **visible lights**, such as headlights in the fog or a "light saber". Finally, lights can produce special effects such as lens flare or the glow of the sun.

There are also two special properties of lighting that are intimately related to their interaction with materials. The first is **global illumination**, also known as radiosity (lower right). This is the property of surfaces which do not produce mirror-like reflections to reflect diffused light onto other objects in the scene. A good example is a beam of sunlight passing through a window into an otherwise dark room. Other objects in the room, except for the beam itself, would still be seen as a result of the sunlight bouncing off the walls and the floor. In the lower image at the right, you can see the tiles in the pedestal's shadow because of light reflected off the wall. Other surfaces in the scene are brightened up by reflections either from the floor or the wall.

The other special lighting effect is **caustics**. This is the modification of light paths by reflective or transparent surfaces affecting the way the light plays onto another surface. Good examples of caustics are how a magnifying lens focuses a beam of sunlight onto a surface or how sunlight reflected from moving water would make a pattern on the hull of a boat.

Light objects

Lights are created and manipulated just like other objects in the Object Manager. When you click on a light's icon in the manager, the light editor pops up in the Attribute Manager in which a multitude of adjustable parameters can be found. There are ten tabs, each containing many settings and parameters. This introduction will get you started with the most important features. More details can be found in the Help files. Like other objects, a light's position and rotation (scale has no influence on a light) can be changed and animated. Lights can, just like any other object, be placed in hierarchies where they inherit motion from their parent. Additionally, any of a light's adjustable parameters can be animated. This includes color, brightness, size of a spotlight's cone, visibility and any of hundreds of other parameters. In the following series of images, the spot angle, color and brightness of the light have been animated with just two keyframes.



Light Object [Light]

General

Color: R 255, G 255, B 255

Intensity: 100 %

Type: Omni

Shadow: None

Visible Light: None

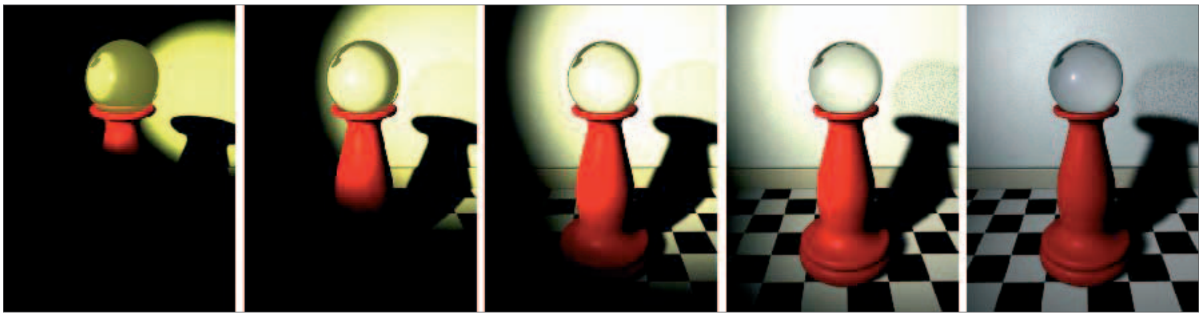
No Illumination ☐ Show Illumination ☑

Ambient Illumination ☐ Show Visible Light ☑

Diffuse ☑ Show Clipping ☑

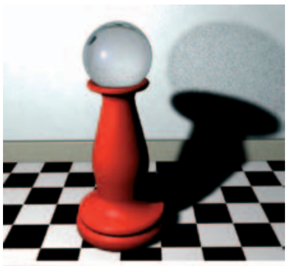
Specular ☑ Separate Pass ☐

Lights may even have materials ("gobos"), including those with moving images. When used this way, the light can be projected through the movie to give a movie theater projection effect.

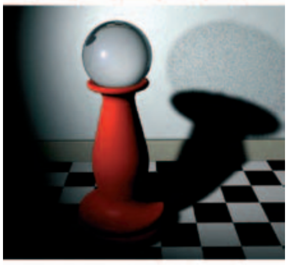


Illumination

The act of emitting light is termed luminance. You should remember the term from our look at materials. When an object is lit from an outside source, it is said to be illuminated. CINEMA 4D offers several light types with different patterns of illumination.

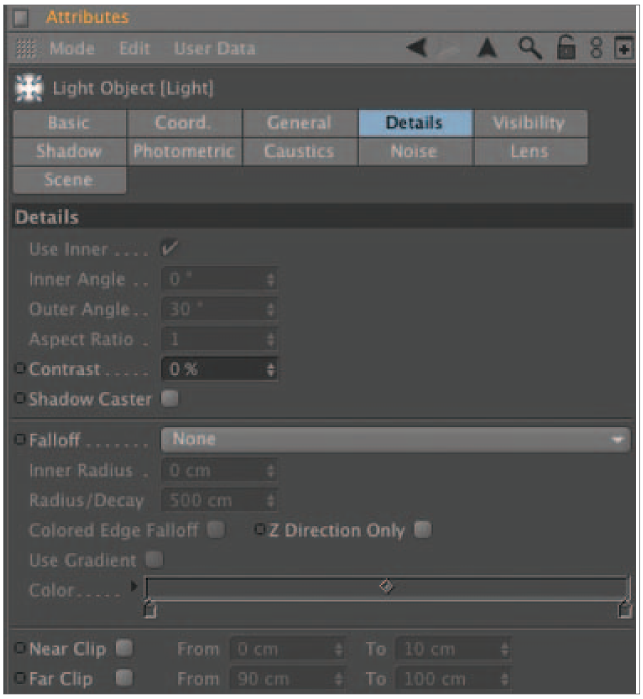


Omni: Omni light is the default light that is created when the light icon is clicked. Omni lights are a good choice for many applications. Omni lights cast light radially in all directions, outward from the origin of the light. Uses range from a light bulb to a star in a science-fiction scene (image top right).



Spot: These cast light in a narrow beam, like the spotlights in a theater production or concert. You have your choice of round or square Spot lights, cone shaped or parallel spots. Natural uses include headlights, theater lights and the glow from a TV screen or a movie being projected (using the square spot). Spot lights can also be used in subtle ways to highlight important features in a scene (image center right).

Infinite: The sun is a good example of an infinite light. These lights cast parallel rays, as if from an infinitely large plane. The illumination is cast along the light's positive Z-axis. Only the direction the light faces matters to the scene (image bottom right). Notice the difference in shadow shape between the Infinite light (radiating from a plane) and the other two types (radiating from a point) listed above.

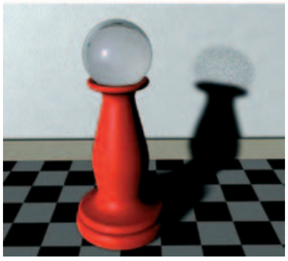


Details

Use Inner . . . . ✓
Inner Angle . . 0 °
Outer Angle . . 30 °
Aspect Ratio . 1
Contrast . . . . . 0 %
Shadow Caster □

Falloff . . . . None
Inner Radius . 0 cm
Radius/Decay 500 cm
Colored Edge Falloff □ Z Direction Only □
Use Gradient □
Color . . . . .

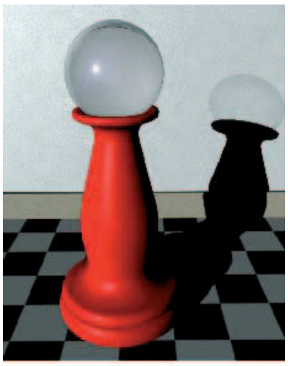
Near Clip □ From 0 cm To 10 cm
Far Clip □ From 90 cm To 100 cm



Area: Area lights are also similar to parallel lights, except that they have a limited size or area. This can be a flat plane from which the light emanates outward or a three-dimensional surface acting as a membrane. However, although the light is cast in the general direction along the Z-axis, it spreads as it travels outward. Area lights can be used to simulate light bouncing off walls (faster than Global Illumination, although not quite as realistic) and can give broader highlights on reflective objects.

Parallel: A parallel light behaves just like a Infinite light except that the illumination begins at the origin of the light itself. In this case the position of the light does matter. Nothing on the back side (negative Z) of the light's XY plane will be illuminated (also, image bottom right).

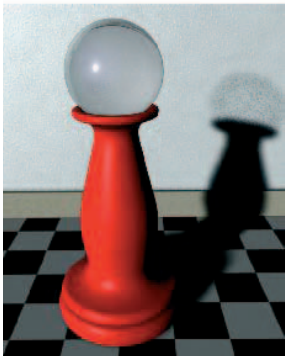
Some features of a light's illumination are set in the General tab of the Attribute Manager. This includes the type of light, its color and the light's intensity. The intensity is set by the *Brightness* slider and may be set to more than 100 if needed.



Most of the illumination parameters are set in the Details Tab. The radius of a spot may be numerically set in a text box as well as interactively in the editor. Lights may be clipped so they begin a minimum distance away from the light and a maximum distance out where they no longer illuminate. One of the most important properties is falloff, which describes how the intensity of a light diminishes as it travels outward from its source.

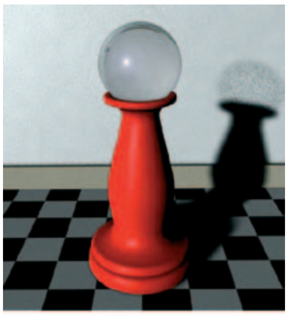
Shadows

Shadows give a sense of realism and help enhance the 3D effect in a scene. CINEMA 4D offers three types of shadows.



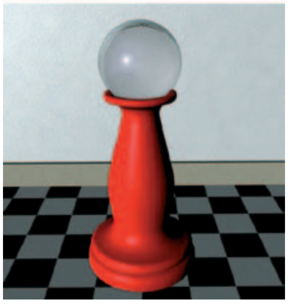
Hard: As the name implies, hard shadows have a hard, defined edge (upper image). Hard shadows render relatively quickly but are a bit too harsh for many scenes in natural settings. Light exhibits a property called diffraction which causes it to bend a bit around edges. Due to this diffraction, most natural shadows have edges which become softer (more diffused or blurry) the further away the shadow is cast from the object. Even though unrealistic, hard shadows work well for many applications.

Soft: Soft shadows produce a more natural diffused edge (lower right), although the effect is uniformly diffuse even close to the object where a shadow would normally become sharper. Soft shadows calculate "shadow maps" to produce their effects, which adds to the memory requirements of a scene. The size of a shadow map can be adjusted to produce finer results at the expense of memory.



Area: Area Shadows are the most natural. They are sharper close to the object and become more diffused further away. However, you should use Area Shadows sparingly since they add substantially to rendering time.

A general rule of thumb is to have only one light in a scene cast shadows. This helps realism by reducing "visual clutter" and can save substantial render time. Of course certain situations, such as stage lighting, warrant multiple shadow-casting lights.

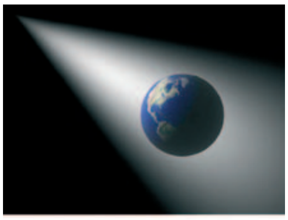


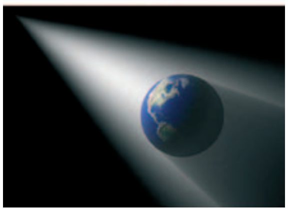
Of course, if a light does not cast shadows, its illumination will pass through objects to illuminate those on the other side, as if the objects were transparent to light. Both figures at the right use the same light, except that shadows have been disabled for the lower image. Notice that the shading on the pedestal appears the same on all but the floor in the bottom image. It appears lit as if the pedestal weren't there. If you do not want a light illuminating objects hidden behind other objects (from the light's point of view), you must activate shadows for that light. Other ways of controlling how light affects objects in the scene include adjusting the illumination fall-off or clipping so the light does not reach more distant objects. For a given light, you can also choose which objects are not to be illuminated by the light (via the Scene tab).

Shadow types are chosen in the General tab and parameters adjusted in the Shadow tab. Some of the adjustable parameters include shadow density and color as well as values that balance a trade-off between quality and render efficiency for soft and Area Shadows. Individual objects may also be given a *Compositing* tag in the Object Manager in which you can determine whether that object receives shadows or casts shadows.

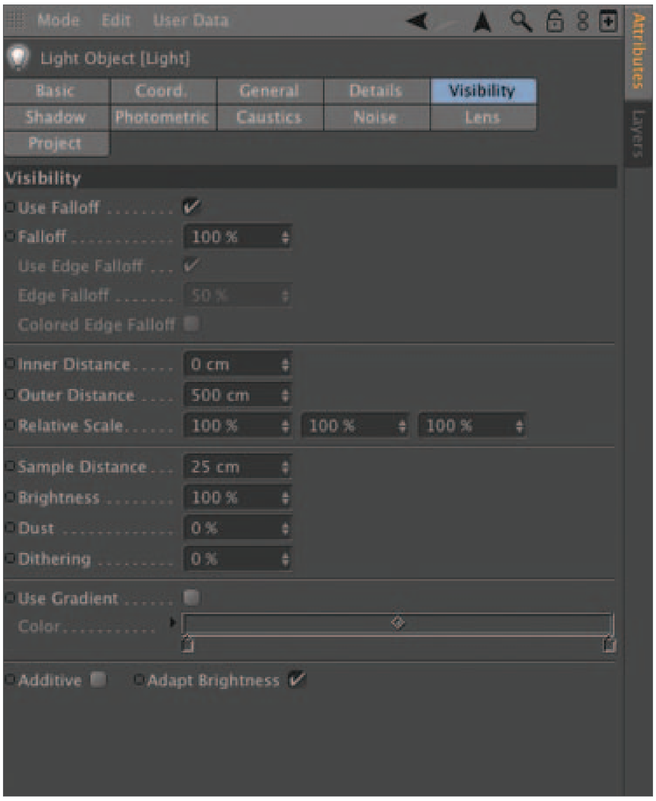
Visible lights

Omni, spots and the *Line* type of Area light lights can not only illuminate other objects, they can illuminate the atmosphere. This is referred to as visible light and can produce effects such as headlights in the fog or a lamp in a smoke-filled room. There are two principal types of visible lights, **visible** and **volumetric**. The principal difference is that *Visible* lights (upper right) pass through objects much like the illumination of a light with shadows off. With *Volumetric* lights (lower right), the visibility is blocked by objects intercepting the visible light rays. Like soft shadows, volumetric lights use shadow maps. They are more computationally intensive and render much more slowly than ordinary visible lights but must be used if you need the look of a shadow cast in a beam of visible light.





The Visibility tab offers control of fall-off. This defines how the intensity of the visible light fades over distance. You may also adjust the brightness of the visible light here independently of the light's illumination (which is set in the General tab). Among other rarely used features is the Dust parameter which actually subtracts light from your beam. Lights with dust are most often used as particles in particle systems simulating smoke and fire.



Light Object [Light]

Visibility

- Use Falloff ☑
- Falloff 100 %
- Use Edge Falloff ☑
- Edge Falloff 50 %
- Colored Edge Falloff □
- Inner Distance 0 cm
- Outer Distance 500 cm
- Relative Scale 100 % 100 % 100 %
- Sample Distance 25 cm
- Brightness 100 %
- Dust 0 %
- Dithering 0 %
- Use Gradient □
- Color
- Additive □
- Adapt Brightness ☑

Some of the parameter settings under the Details Tab also affect visible lights such as clipping (shown middle right). This is useful, for example, if creating a searchlight whose beam begins as a sizable disk instead of a small point.

Other effects

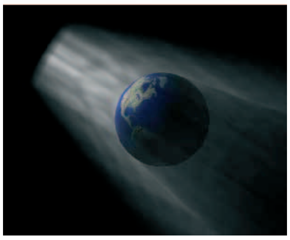
Lights, as mentioned previously, can have materials assigned to them. When this happens, the material acts as a gel which in turn acts like a transparency to give a slide show effect as the light passes through. The effects can range from simulating sunlight passing through louvered blinds to a movie being projected onto a screen.

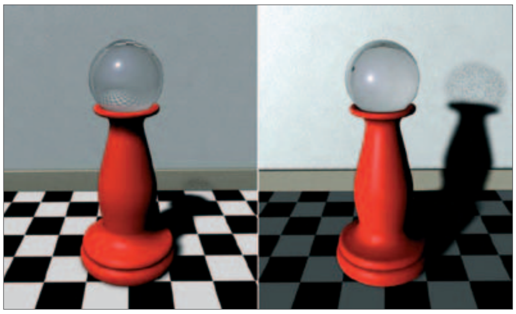
The Attribute Manager has a tab named Noise which adds animated turbulence to the light's illumination or visibility or both (lower image on right). This can be used in subtle ways to show, for example, the slow movement of fog.

Lights may also have lens and glow effects that have their own tab in the Attribute Manager. These can be used to simulate lens flares (which good cinematographers try to avoid) or those cheesy, starlight twinkle effects. They can also be used to produce certain sun and star effects.

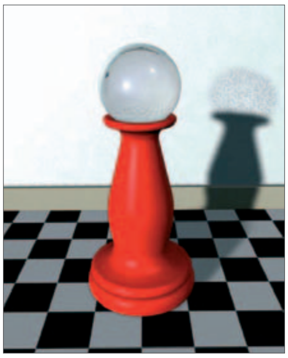
Using lights

Good lighting is very important and can be a daunting task in complex scenes. Lights interact with each other making trial and error a necessary part of the lighting process. For most purposes we can group lights, by their usage, into four categories.





Key light: Almost every scene has one key light. This is the brightest, most dominant light that casts the shadows and sets the overall tone in the scene. Outdoors, of course, this would be the sun. In your dining area, in the evening, the ceiling light over the table would be the key light. The brightness, color and angle from which the key illuminates everything and gives the scene its mood. Lower lights produce longer shadows resulting in a more somber, dramatic mood than lights that are positioned higher. Lights shinning directly upward under people's chins can result in a sinister effect.



Fill lights (global): Illumination from just one light is too harsh and unnatural for most applications so fill lights are used to simulate some of the Ambient light from the sky or bounced off other surfaces like walls in a room. Notice how the use of one carefully placed fill light in the image at the right has given the scene a more realistic appearance (a single light with Global Illumination can also produce this look but with greatly increased render times). Generally one or two fill lights are used to bring out detail which would otherwise be lost in the shadows of a single, harsh key light. Even with science-fiction shots in deep space, where there is no atmosphere to cause dispersion of light, a fill light is used. Otherwise the scene will appear too harshly lit (although harsh lighting is more realistic in space).

Fill lights (local): Sometimes you want to emphasize or draw attention to some local feature of the scene. For this, a spot trained on a narrow region can be used to subtly highlight a certain character in a group, bring out an important detail otherwise lost in shadow or lend a bit of color to part of a surface like a blush on a cheek.

Practical lights: These are lights that are actually seen in a scene. A table lamp, car headlight, theater spotlight, rocket engine, TV screen and much more all are examples of practical lights. These may be used for their illumination (a table lamp), for their illuminating nature (car headlight in the fog) or for both (theater spotlight in a dusty theater).

Lighting can get quite complicated. It is possible, and sometimes necessary, to use several light objects to create the effect of one light. For example, a table lamp's illumination might be too bright to properly cast the subtle shadows made by the lamp shade. For this you might use a separate light at the same position to cast shadows. You would probably also use multiple lights to create a sun for a space scene.

Lights can also be animated. Just about any parameter can be animated. This includes obviously useful functions like dimming the illumination and changing the color of the sun as it sets. However, even parameters such as shadow density, falloff (both of visibility and illumination) and much more can be animated to produce special effects. In fact, many of the dozen or so parameters which appear in the Attribute Manager for a given light can be animated either in the Timeline or directly in the Attribute Manager.

Although lights have many parameters, interactions, uses and complications, you will see in an upcoming tutorial that it is relatively easy to get started with a three-light system that works quite well for many scenes and serves as a starting point for others. So, let there be light!

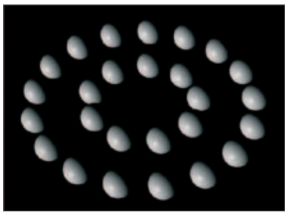
Lighting Intro Tutorial

This tutorial follows the Introduction to *Lighting* handout. It serves as an introduction to types of lights and some of the more important parameters associated with them. Since you will render often in this tutorial, you should hide elements in both the editor and the renderer when asked to hide an object.

1. Types of lights

CINEMA 4D offers several types of lights, each with specific properties and applications. In this section we will look at how each of the light types illuminates the surface of objects. Later on we will investigate other differences between the types of lights including how some lights can be made visible where others can only illuminate surfaces.

Open the file "Lighting_Tut.c4d" The object Light Types should be visible and the other objects should be hidden. The camera Cam : types should be selected as the Scene Camera for the viewport with just a single viewport showing. Expand Types and the Child object Lights. Unhide and select Infinite light and observe the results. Render.



Before you unhid the light, the scene was lit by the *Auto Light* by default. As soon as any other light is activated, the *Auto Light* will be turned off. Notice that the distance light, although in the center of the array of spheres, illuminates the entire array from the same direction along the light's positive Z-axis.

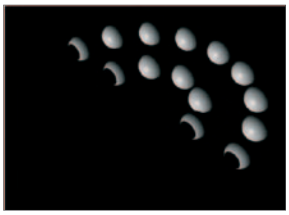
Try translating the Infinite light object around in the view.

Nothing should change. A Infinite light always radiates from an infinite distance, with the rays all parallel to the Z-axis. It is not affected by position, only the direction of the Z-axis.

Return the light to the origin (undo). Rotate the light about the heading.

The illumination on the spheres should change to reflect the new direction of the light rays.

Return the heading to 0º and hide the Infinite light. Unhide and select parallel Light. Render.

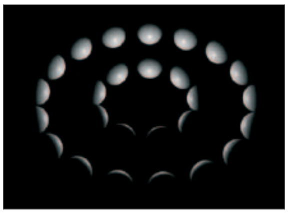


The effect is the same, except that the rays begin on the XY plane of the light object and travel only in the positive Z direction. Anything in the scene behind the light (toward the minus Z) will not be illuminated by the light.

Move the parallel light object along its Z-axis and render to observe the changes to the scene.

Rotation of the parallel light will have the same affect as rotating the Infinite light, except that, as the XY plane rotates, a different portion of the scene (being the plane) will be hidden to the light. Try it if you need to clarify.

Return the light to its original position and orientation and hide it. Unhide and select omni Light. Render.



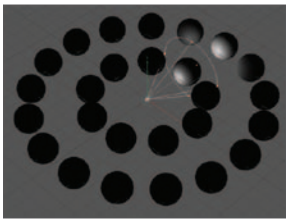
This should appear substantially different. All spheres are now illuminated from a point in the center, the origin of the omni light.

Try rotating the omni light. Undo, then translate it upward along the Y-axis. Render.

Rotation should have no effect, just like viewing a bare light bulb from different angles. However, raising it above the array should change the lighting effect.

Return the light to its original position and hide it. Unhide and select spot Light.

You can see the spot's cone of illumination in the editor window.



Rotate the heading of the spot light.

The illumination is always within the cone whose center is directed along the light's positive Z-axis. There is also a square Spot that emits a pyramid-shaped beam of light. This is useful for TV and movie projector applications.

The two parallel spots behave like the parallel light you investigated above, except that the light, instead of being emitted from the entire XY plane of the light, comes only from a within a circle or square on the XY plane.

Return the spot light to its original position. Change spot Light's Type to Parallel Spot and observe the look of the object in the view.

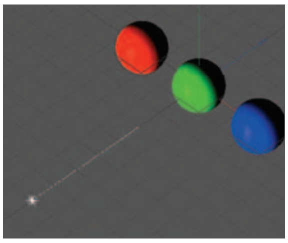
Let's get a preview of a spot's parameters before moving on.

Return the Type to Spot with an undo. Activate the Details Tab in the Attribute Manager. In the viewport, grab one of the four points around the circumference of the circle at the end of the spot's cone and drag outward while observing the value of Outer Angle in the Attribute Manager. Render with a new value of Outer Angle greater than 120°.

Like primitive objects, many light and camera parameters are adjustable not only in the Attribute Manager However, as you just observed, interactively in the editor.

Undo the angle change and hide the spot light.

2. Light color



In this section we will visit the General tab of the light editor and change the color of a light.

Switch the view camera to Cam : color. Hide Types object. Unhide and unfold Color.

Light is a Infinite light. Although its position makes no difference, it has been placed so that we can more easily visualize the illumination coming from it. The three spheres are colored in the primary colors (R,G,B) and the light is white. Predict what will happen if you change the light's color to red.

Select Light in the Object Manager. Click on the General tab in the Attribute Manager (if not already open by default). Change the color to red (HSV = 0, 100, 100).

Are the results as predicted? Since there is no blue or green in red light, there is none to be reflected by the blue or green objects. Since they reflect no color, they appear black.

The lesson here is that the surface color of your object is a function both of the material applied to the object and the lights illuminating the object. Even a small amount of color can have a subtle but noticeable effect on the mood of a scene.

Change the color of the light to yellow (60, 100, 100).

Because yellow is a combination of red and green light, both the red and green spheres appear normal. Yellow is also the color you get when you take the blue out of white light, therefore the blue sphere appears black (you may have been taught that yellow and blue make green. This is pretty much true with paint that *subtracts* color. However, when yellow light is added to green light, as happens in TVs, computer monitors and your eye, the result is white. This is the *additive* method of combining color).

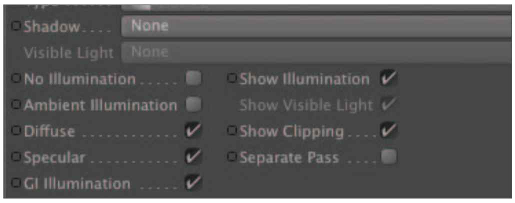
Reset the light's color to white. Change the brightness to about 50 by adjusting the Brightness slider in the color panel of the light editor.

No mystery here. Most of your fill lights will be at this level or less. Sometimes the key light or certain practical lights, such as car headlights or the tractor beam from the mother ship, need to be made quite a bit brighter.

Set the value in the Brightness text box to 300.

If that is not enough you can enter values up to 1000%. You will also notice that the General tab is where you set the light type - the type of shadows and whether the light is visible.

The three "Show" check boxes in the lower right only affect whether certain characteristics, such as the cone angle of a spot, are shown in the editor. They have nothing to do with the appearance of the rendered image. They only reveal handles that allow interactive adjustment in the editor.



○ Shadow .... None

Visible Light None

○ No Illumination ..... ☐ ○ Show Illumination ☑

○ Ambient Illumination ☐ Show Visible Light ☑

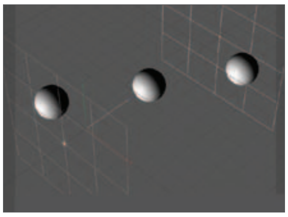
○ Diffuse .... ☑ ○ Show Clipping .... ☑

○ Specular .... ☑ ○ Separate Pass .... ☐

○ GI Illumination ..... ☑

The *No Illumination* is used when you want certain other effects from the light ,such as visibility or lens flares, without having the light actually illuminate objects.

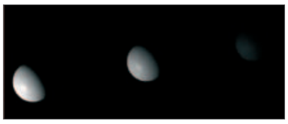
3. Falloff, clipping and cone angle



With most real light sources, light intensity diminishes at greater distances from the light. The difference in distance to the sun is insignificant for sunlight at various points on the Earth's surface so the intensity remains constant. However, a flashlight beam gets considerably weaker after just a few meters. This diminishing of intensity is called falloff and is controlled in the Details Tab of the light editor.

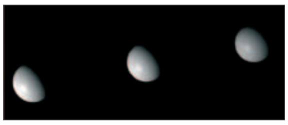
Switch to the camera Cam : falloff. Hide Color and unhide Falloff. Select the object Light under Falloff.

Notice the relative positions of the light and the three spheres. The parallel light is represented by the plane containing the light's object axes. The other plane, outward along the Z-axis, shows the falloff distance.



With Light selected, open the Detail tab and make note of the Radius/Decay. Render the view in the editor.

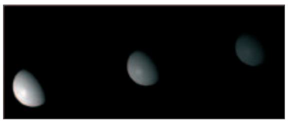
The falloff distance (*radius/Decay*) is the point at which the illumination is reduced to zero. In this case the falloff distance was 1300 and the farthest sphere was about 1200 away from the light source. You need to look carefully at the rendered view to see the farthest sphere barely picking up the last bit of light before the illumination totally fades out.



In the editor, interactively move the falloff distance farther away to about 2500. Watch as the value changes in the Attribute Manager to gage when to stop. Render.

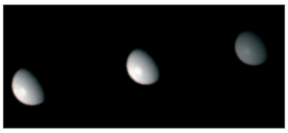
Now the farthest sphere is about halfway to where the light fades to zero and, indeed, appears about half as bright as the first sphere closest to the light. Take a close look, we are going to change the way the light falls off.

Change the Falloff type from Linear to Inv. Limited. Render.



Now the light falls off faster and faster the farther away it is cast (on a graph, the plot of light intensity vs. distance for this inverse falloff would look like half a hyperbola). If you want to accurately depict falloff from a point source such as a spherical light bulb you would, at least according to the laws of physics, use the inverse square mode that falls off even faster.

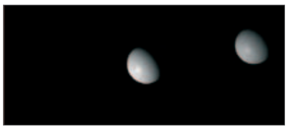
All of this has assumed we want our falloff to begin immediately at the light source. In most cases this is probably true but you may choose to have the light radiate out a certain distance (*inner Radius*) before beginning to falloff.



Set the Inner Radius to 700 and render.

Notice you can also set this interactively. You should see that the middle sphere is about as bright as the first because the falloff does not begin until the second sphere.

Reset the Inner Radius to 0, the Radius/Decay to 2500 and the Falloff type to linear. Render.



Sometimes you do not want a light to begin right at the source or you want it to stop precisely at a certain distance so it does not illuminate a particular object. This is called clipping. Let's adjust the clipping so the light does not illuminate the closest sphere.

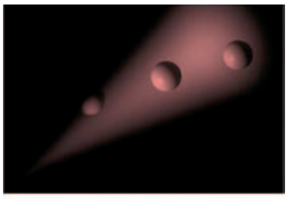
Still on the Detail tab, check the Near Clip box and set both From and to values to 400. Render.

Our closest sphere is not lit because the light is clipped (cut off) to a distance of 400. Clipping also affects visibility of lights, as discussed in the next section.

See if you can cause the light to illuminate only the center sphere.

The other important parameter in the Details Tab is the *Outer Angle* (grayed out now because it only applies to spots). This refers to the angle of the cone of radiation for spot lights. You have already played with this interactively. In the Attribute Manager you can set the value numerically. We will take a look at the *Inner Angle* parameter in conjunction with shadows later.

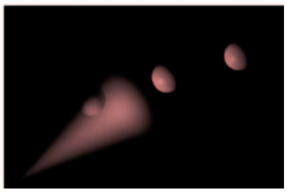
4. Visible lights



Visible lights are used when you are interested in seeing the light beam itself like street lights on a foggy evening.

Switch to the camera Cam : visible. Hide Falloff, unhide and unfold Visible. Render. Select Light and, in its General tab, select Visible for the Visible Light type. Render.

Kind of foggy out tonight.



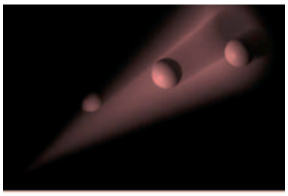
Open the Visibility tab of this light. Change the Outer Distance to 1000 and render.

In both cases the visible light emitted to the distance specified in *Outer Distance* and stopped without diminishing along the way. Notice that the visible light has no effect on the illumination of the spheres, only the visible light cone was shortened.

Return the Outer Distance value to 2000. Check the Use Falloff box leaving the value at the default of 100. Render.

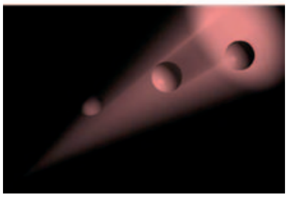
The 100 in the Falloff box means that the visible light diminishes from its peak value (set by a combination of *Brightness* in the General tab and *Brightness* in the Visibility tab) down to zero between the *Inner Distance* and *Outer Distance* values. From its origin to the *Inner Distance*, the visible light will remain at peak value.

You may have noticed that the visible light seems to travel right through the spheres and can be seen in the region behind the spheres which should be in shadow. If this is not desirable, you must use volumetric lighting.



Deactivate falloff. In the General tab, set the Visible Light type to Volumetric. Note the advisory text at the bottom of the editor where it says Render Time : High. Render.

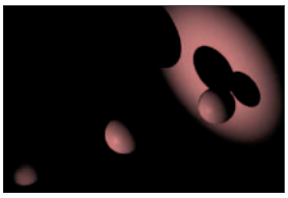
That certainly looks more like a headlight in the fog, but you should definitely take the slower rendering speed into consideration. In a complex scene this can be excruciating so get by with visible vs. volumetric lighting when you can.



Unhide Plane and render.

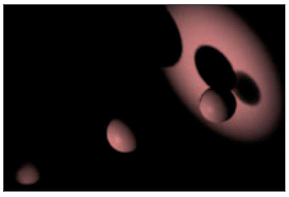
Why do the spheres seem to block the visible light as it travels outward but the plane remains lit as if the spheres weren't there? The plane is not illuminated by the "visible light" aspect that gives the foggy appearance. Like all lights, this one needs shadows if should not shine through objects.

5. Shadows



As we proceed through our brief overview of the shadow types, pay attention to the advisories in the General tab of the light editor as to the *Memory Requirement* and especially the *Render Time*.

Switch to the camera Cam : shadow. In the General tab, change the Visible Light to None and the Shadow type to Hard. Render.



That looks better. Suppose you want a bit of illumination in the shadow areas.

In the Shadow tab, change the Density value to 50%. Render.

Hard shadows render quickly and take up little space. However, in reality a process called diffraction causes light to bend around edges, making shadows less than sharp. You can simulate this using soft shadows.

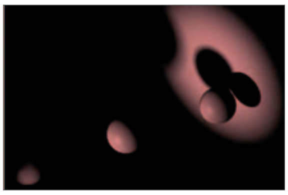
Reset the shadow density to 100. In the General tab select Soft as the shadow type. Render.

This is probably more realistic, depending on what effect you are after. You can get more detail in soft shadows by increasing the shadow map size to the detriment of, you guessed it, render speed and memory. You can also adjust the softness of the shadow using the *Sample Radius* parameter.

Adjust the Sample Radius to 10 and render.

Slower? There is still one problem with our shadow, at least in terms of ideally imitating nature. Objects closer to the surface receiving the shadows should be sharper than those further away. If you really need this perfection and have time for rendering, use Area Shadows.

Back in the General tab, change the shadow type to *Area* and render.

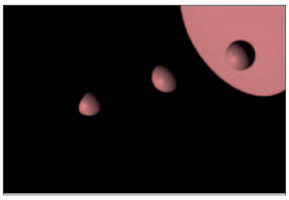


It probably took a long time to render only four objects, no textures, no antialiasing. Imagine rendering a major scene with Area Shadows. Again, 3D is full of decisions to make. If you are rendering a photorealistic still and have all night to let the computer run, go for it. If You are rendering 600 frames with many objects and textures, you may want to reconsider...

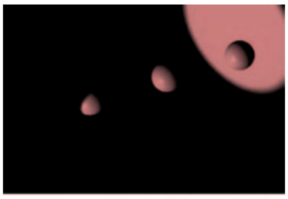
6. Edge falloff

You have seen how to control the falloff of both intensity and visibility of lights as the light emits farther from the source. Spot lights also allow you to specify how the light varies from the center to the outside of the cone.

Switch back to the camera Cam : visible. Turn off the shadows (none) for the light. In the Details Tab, uncheck Use Inner and render.

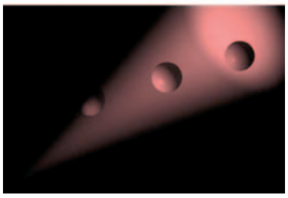


When *Use Inner* is enabled, the value entered in *Inner Angle* specifies the angle from the light's Z-axis toward the edges of the cone, at which point the light is still at maximum intensity. Beyond that angle it begins its falloff to zero. A value of zero means the light is its brightest at the center and its falloff begins immediately. A value equal to that of the *Outer Angle* results in a hard edge with no falloff, the same as unchecking the *Use Inner* box.



Assuming Outer Angle is still set to 30, recheck Inner Angle and set its value to 25. Render.

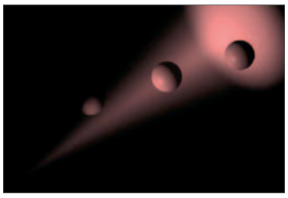
The light illuminating the plane should appear uniform, except where it fades near the edge. Let's see how edge falloff works with visible lights.



Back in the General tab, activate Visible light. In the Visibility tab, make sure Use Falloff is unchecked and that Use Edge Falloff is enabled. Set the value of Edge Falloff to 0 and render.

The visible light shows at maximum intensity from one edge of the cone to the other. This is the same as unchecking *Use Edge Falloff*.

Set the **Edge Falloff** to 100 and render.



Now you see the strength of the visible light fading from maximum intensity down the center to zero at the edges of the cone. To summarize: You set Edge Falloff for the illumination in the Details Tab and the visible light edge falloff is set in the Visibility tab.

There are dozens of additional parameters and features of lights. You may find occasion to use some of these in your projects. You can find more information in the Help files. However, the basic tools covered in this tutorial will provide you with the lighting you need for the bulk of your work.

Composing and Lighting a Scene

In this tutorial you will compose a simple scene and light it. The lighting system you will be introduced to is a simple three-light system which is satisfactory for many scenes and can serve as a starting point for many others. You will also use "practical lights" to light your lamp.

1. Merging scenes

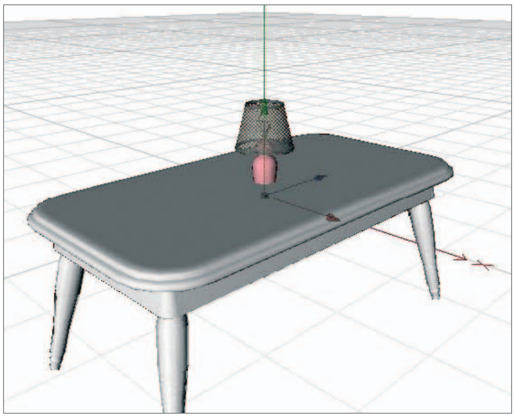
We will put together our wooden table along with our lamp, vase, a wall and a floor. This will also give you practice in modeling and texturing.

Create a new folder within your student folder named "Table Scene". Open a new CINEMA 4D file and save it in the new folder as "Table Scene 1".

You are going to bring the table and lamp into this scene. There are a couple of ways to do this. First we will bring in the table by copying and pasting.

Open the file "Table 2.c4d" and select the final, textured version of your table. Copy this object to the clipboard using Cmd+C. Return to the current scene by going to the main menu and selecting Window / Table Scene1. Paste the table into this scene with Cmd+V. Close the file "Table 2.c4d".

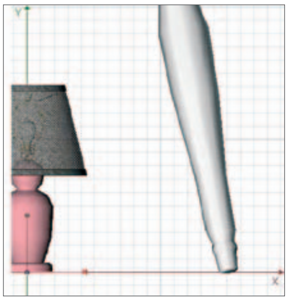
You will notice that all of the textures for your table were copied and pasted in addition to the objects. Copy-and-paste is good way to bring in one element from a scene when you do not need the rest (such as our different versions of the top in the "Table 2" file). Let's bring the lamp into the scene with another method.



From the main menu, choose File / Merge.... in the dialog, navigate to your "Table Lamp 2" file and Open it.

The lamp, along with its materials, should now be in the new scene with your table. Your scene should look something like that at the right. The lamp looks too small for the table and needs to be repositioned. First, we will set the editor camera to a focal length that is appropriate for the scene - you can tweak it later if you like.

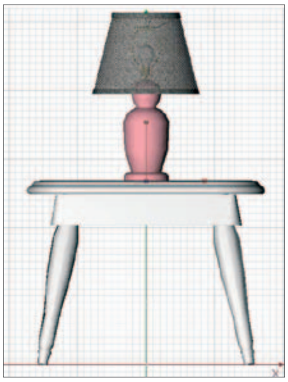
Set the Focus of the camera to 45.



This is a bit wider than normal (about 55) which is good for indoor scenes where the space is cramped. Since this scene will have a definite floor, let's reposition the table to sit on the floor before moving the lamp.

In the front view, move the table up until its legs just touch the Z-axis (at Y = 0). Move in for a more precise final adjustment.

Notice that, due to the angle of our legs, they only touch the floor at a small point. If you want, you can lower the table into so the outside edge of the leg bottom makes contact with the floor. If the process of moving the objects is very slow you may wish to increase speed in the editor by lowering the display resolution. Do this in the view menu: Options / *Level of Detail Medium*.

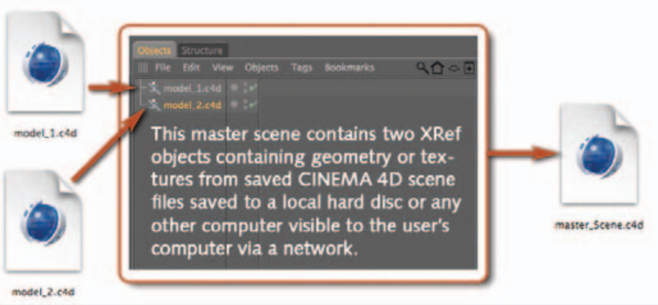


While still in the front view, reposition the lamp to sit on the table's top. Using the Model mode, scale the lamp until it looks properly proportioned to the table.

Since you are still modeling, use the Model mode to scale the lamp (this will change the size of the lamp without distorting its axes ,which would occur if the Object mode were used. Although it will not make a difference here, it is still good practice. If you were to subsequently add a Child object to the lamp, after scaling with the Object mode, any movement of that child could lead to unwanted results due to the distorted axis system of the Parent Lamp).

Let's add one of the vases you modeled earlier to the table by using an XRef Object.

The XRef Object is similar to an instance object but works over multiple scenes allowing individual CINEMA 4D "reference scenes" to be combined into a "master scene" and rendered there as if all objects were part of the master scene.



This master scene contains two XRef objects containing geometry or textures from saved CINEMA 4D scene files saved to a local hard disc or any other computer visible to the user's computer via a network.

One example of its use would be if you were working with a very large scene that contained several complex objects. One person would be responsible for placing or animating the objects and others would create the actual models in their own individual scene files. Objects that are part of individual reference scenes can be added to the master file via XRef. One of the most powerful features of the XRef Object is that it enables the master scene to be updated automatically whenever a reference scene is modified.

From the main menu, choose Create / XRef / Add XRef... to create an XRef Object. You will be prompted to choose a CINEMA 4D file. Navigate to your "Vases 1" file and Open it.

If you look in the Object Manager you will see that the name of the XRef Object is Vases. An XRef Object will always adopt the name of the reference file you have chosen. This is one occasion where you will not be asked to name the object in your scene, as CINEMA 4D will do this for you.

In the Object Manager you can see the XRef Object 'Vases' and a list of objects contained in Vases 1.c4d.

The contents of Vases_1 behave almost exactly as any other object in your scene. They can be translated independently, animated and manipulated in may ways, however they do have limitations. You will notice, for example, that textures applied to objects within an XRef cannot be edited in the Master file. In order to change parameters such as this we must open the original file and make our changes there.

Importantly an XRef component object's visibility can be switched on and off using the standard method in the Object Manager. In this way you can choose which of the two vases you saved in your "Vases 1" file will be displayed in the current scene.

Position and scale Vases 1.c4d so it is in proportion to your lamp and table. Using the top and perspective views, reposition the lamp and vase on the table (Y-axis locked). Texture Vases 1.c4d with a new material of your choice.

In many ways XRef Objects behave the same as normal objects. For example, XRef Objects can have materials assigned to them or they can be deformed using Deformer objects (Shear, Bend, etc).

If you have time, try to create a nice material using the regular Material Editor. The volume shaders are nice but slow to render and you have already used several of them in your scene. Try some of the channel shaders. Try using *Lumas* in the luminance channel and *Noise / "Cranal"* in the bump and diffusion channels to give it a wild look (for *"Cranal"*: up the *Low Clip*, lower the *High Clip*, increase the *Brightness* and *Contrast* and reduce the *Global Scale* substantially. Also reduce the bump *Strength*).

2. Adding a floor

Now we will create a tiled floor.

Hide the lamp and Vases 1 but let the table show so we get a sense of scale while working on the floor.

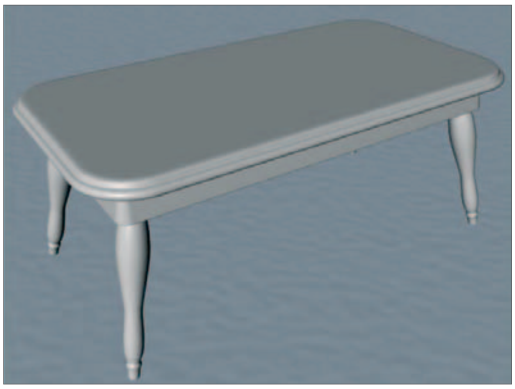
If you are working on a really fast system, you can skip the following step.

Create a new material, leave everything at the default settings and apply it to each of the table components currently having a Texture Tag (but not by dragging over the existing tags).

This will temporarily cover the table's grain and greatly speed up rendering while we work on the floor while still allowing use of the table as a scale reference object.

While in the perspective view, create a Floor Object.

This will be positioned, by default, at Y = 0, which is where we want it. It will extend indefinitely on the XZ plane to the horizon, but we will only see what the wall allows. Now we need to create materials for our floor. You will notice there is a *Tile* surface built into CINEMA 4D, but we will not use it here. You cannot get the grout lines narrow enough for a standard look and they seem to get larger as they go off into the distance. However, as the old saying goes, there is more than one way to texture a surface. Creativity in 3D does not just mean being able to realize visual ideas but also being able to get the results you want from the tools you have. We will create one material for the tiles and a second one, that will overlay the tiles, for the grout.



Create a new material named floor. Apply the material to the floor object. The default projection will be Flat which is what you want for a floor.

If you select the *Texture Axis* mode and inspect the Coordinates Manager, or look at the Coord Tab of the Attribute Manager, you will notice that the texture is rotated -90° around the pitch axis. This makes the Z-axis of the texture perpendicular to the plane of the floor which is what you need since textures are projected along the Z-axis when flat mapped.

Set the color of the floor material to some relatively light color Activate the Bump channel and select the Noise shader. Edit the shader, changing the Space to UV (2D). Back in the main bump channel, set the Strength to 1 (or less). Render a portion to see the effects.

The bump will give the tiles a subtle wavy surface, characteristic of many types of tile. We should typically add diffusion to a surface such as this but the bump will probably add enough variation to lose the clean computer look. Using the 2D version of the *Noise* shader will speed up rendering.

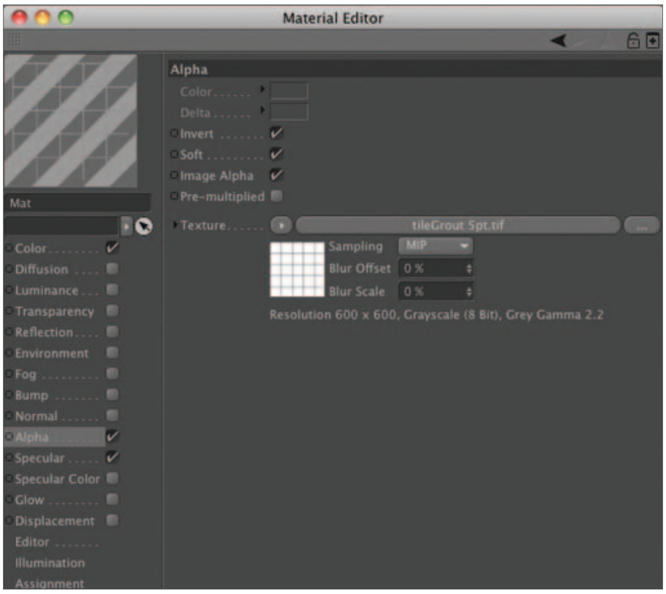
In the material's specular channel, set the Width to 60 and the Height to 60. Render a test area. Reduce the bump's Strength if necessary.

Wait until after placement of the lights to make any more adjustments to these values. Now you will create the grout that will define the edges of the tiles.

Create a new material named grout. Change the color to some color which is contrasting to, but compatible with, your floor color. Apply the material to your floor. Again, the default values are what you want. Render.

Since the grout Texture Tag is farthest right of the two Texture Tags associated with the floor object, it is topmost and hides the floor texture underneath. We will use an alpha channel to get rid of the grout texture at the locations we want tiles. That is, our alpha channel will define the grout pattern. Let's set up our texture map first.

The texture you want is titled "tileGrout 5pt.tif" in the "tile grout lines" folder within the "tex" folder of your student folder. Create a new "tex" folder within the "Table Scene" folder and drag a copy of "tileGrout 5pt.tif" into it.



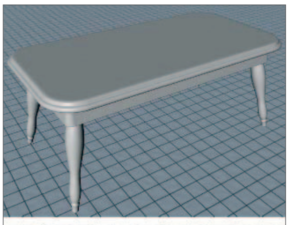
Alpha

Color
Delta
Invert ✓
Soft ✓
Image Alpha ✓
Pre-multiplied □
Texture tileGrout 5pt.tif
Sampling MIP
Blur Offset 0 %
Blur Scale 0 %
Resolution 600 x 600, Grayscale (8 Bit), Grey Gamma 2.2

Remember, you need to have copies of texture images (bitmaps) within a "tex" folder within the folder containing the CINEMA 4D file. If you are going to use the canvas lamp shade you will also want a copy of "Canvas.tif" in the tex folder.

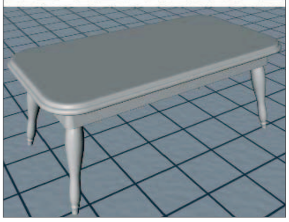
Activate the alpha channel of grout and import the image "tileGrout5pt" (from the proper tex folder, of course). Disable the Image Alpha box, make sure Soft and Invert are checked. Render.

We now have tiles. We left the table visible for a reference to adjust the size of the tiles. Right now, they appear too small for a typical 12-inch tile.



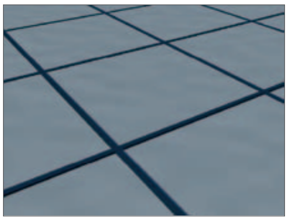
In the Tag tab of the Attribute Manager, change the Length values both to 300 and render.

If you were using shaders in the material you would need to increase the *Size* values in the Coord Tab instead. If you think the tiles should be a different size, re scale them. Our grout needs to be recessed for a more realistic look.



Copy grout's Alpha channel texture into the Bump channel. Leave the bump's Strength at 20. Render.

Grout is typically much rougher than ceramic tile. You will need to adjust the specular values accordingly.

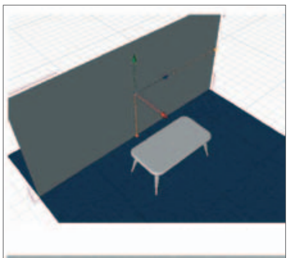


Set the specular Width to 40 and the Height to 30. Render.

This should be looking like a reasonable representation of tile by now. You could go on to add graininess to the grout but, unless your camera is going to take a slow dive toward the floor, this will probably not be necessary. On the other hand, you might want to increase detail if you are rendering a 2048 pixel wide still for presentation purposes. Unless something really says "change me", reserve any more modifications until after setting up the lights.

3. Adding a wall

We will create one vertical wall behind the table.



Create a Plane primitive and name it "Wall". Set its Width to 1000, its Height to 2000 and its Orientation to the positive X direction. Set the Segments values both to 1.

If necessary, we can adjust the size of the wall once we have placed the camera. Since the plane will not be deformed in any way, reducing the segments reduces the polygon count and you know what that means.

Back up in the view and position the wall an appropriate distance to the far side of the table and raise it up so only a portion is still below the floor.

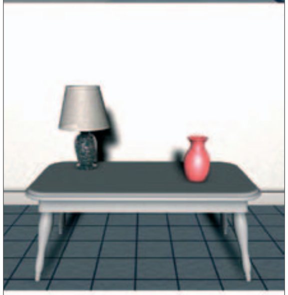
Let's add a texture to the wall. Let's go for some realism with a neutral color and some drywall texturing.

Texture the wall! Use Noise in the bump channel and experiment with different types like Nutous. Use flat projection with the Z-axis of the texture pointing into (or out of) the wall.

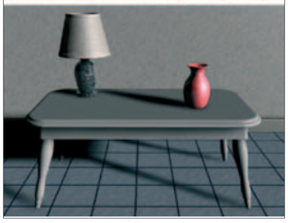
A couple of hints to get you started: You should use flat projection. However, it will not be oriented properly. Flat projection is done along the Z-axis (of the texture) so you will have to rotate the texture's axes. A good stucco bump is the *Noise* shader with the *Nutous* pattern. Do not forget some diffusion (set to a larger scale than your bump).

Make a baseboard for your wall. Make the baseboard a Child of the wall and position it.

Baseboards are usually finished with glossier paint than walls. You can create an oak baseboard but it will increase render time. Also, do not simply extrude a rectangle or you will get that too-sharp computer edge on your baseboard. In the figure at the right is a detail showing the wall texture and the subtle use of diffusion on the baseboard.



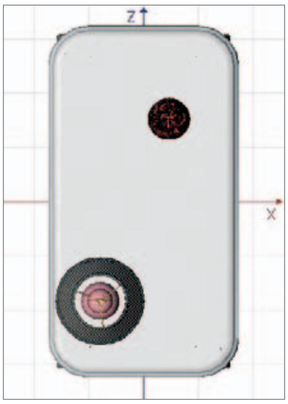
4. Setting up the shot



Here is where you arrange the objects and define how the scene will be viewed. You also need to have some idea of your lighting arrangement to get the best effect. For example, maybe we want the key light to be sun coming in through a window. If we place the sun behind our (the camera's) shoulder and look straight at the wall, the lighting will be "flat". This means that the three-dimensional nature of objects in the scene is subdued and the scene will appear rather boring (upper right).

On the other hand, if we swing the light around more to the side of the camera's view, the scene will take on a more interesting and three-dimensional look. So let's plan to put our key light (sun) to the right (lower right).

Arrange your lamp and vase somewhat like that to the right.



Our table, floor and wall will also look more interesting and three-dimensional from an angle. We are going to swing the camera to the left. This will cause the scene to be somewhat backlit which is good in this case because it will help accentuate our table lamp when we finally turn it on. We also want to be up just a bit above the top so we get a view of the table's grain.

Move in the perspective view until your scene is similar to that at the right. Create a camera object and add a Protection Tag to it.

If you want to later tweak the camera's position, drag the tag onto a Null Object (or any other object), make your changes and drag the tag back.

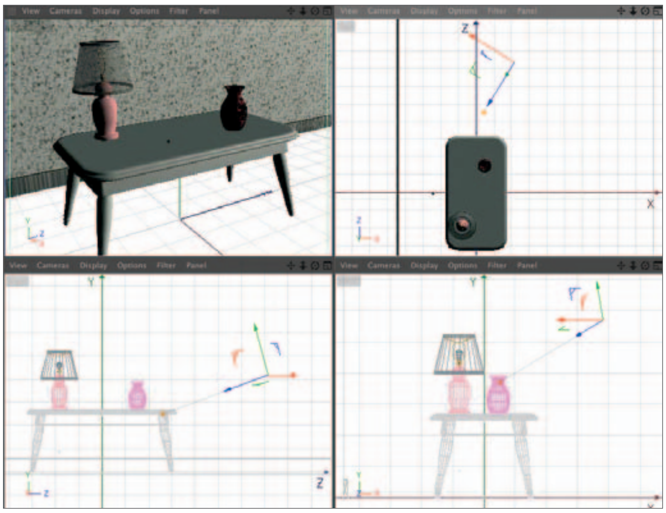
Drag the wall along its Z-axis until it fills the scene (or at least the part that will be rendered).

Once we have finished our lighting setup, feel free to experiment with different viewpoints and focal lengths.

5. Lighting the Scene

As mentioned earlier, we are going to use a simple three-light system, not including the lamp. The three-light system and its variants are the bread and butter of lighting.

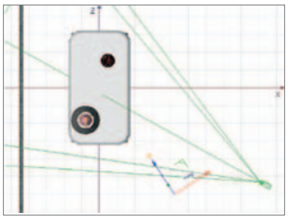
In a three-light system, there is one principal light called the key. The key is the brightest and generally the only light casting shadows in a scene. It usually corresponds to some source that would be related to the scene. For example, in our scene, the key light might be a ceiling light and be placed overhead (night scene). It could also be sunlight shining through a window (day scene). In this case let's use the sunlight. It will be at more of an angle highlighting the texture of our floor and giving us more dramatic shadows.



Turn off antialiasing (in the Render Settings) if it is not already. Create a light object and name it "Key". Make the light Type a Infinite light and change its color to an afternoon shade of sunlight, maybe 60, 10, 100.

Since the light was created at the World origin, we need to reposition it.

Hide the floor and switch to the 4-view mode. In the top view, position the Key as shown with its Z-axis pointing toward the center of the table and the Z-axis making an angle of about 30° with the wall. From the side view, direct the light down onto the surface of the table as shown, also at about a 20 to 30° angle.



Remember, being a Infinite light we really didn't need to reposition it. Only the direction of the Z-axis matters. However, it helps conceptually to place it up and to the side where the window would be.

Unhide the floor and render (in the editor) from your camera's view.

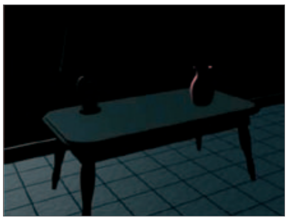
Throughout the rest of the lighting process you will need to switch between the four-view for manipulating lights and the perspective view for seeing the results in a render (you can also render the active window in the four view mode, it will just give you a smaller rendering).

Create a second light naming it "Fill 1". Make this a Infinite light, add a slight bit of blue (190, 8, 100) and set the Brightness to 40. Place this light up where the ceiling would be and to the left of the camera. Orient it so it is directed down and toward the center.

This positioning will give a general impression of indirect light scattered from various room surfaces and will help fill in the shadows of the key light for more realism. Note that the light is not directly overhead and is positioned to make sure the front of objects are not too dark.

Hide the key light and render using only the light from Fill 1.

It is always a good idea to know how each of your lights contributes to the scene. For our third light, we have a couple of options. One is to use a spot and aim it at an object we want to accentuate in the scene. Since we do not really have a featured object and the lamp will also be adding light to the scene, let's select a different Infinite light to simulate reflection off the back wall.



Create a third light, "Fill 2". Position it just in front of the wall and orient it to point straight-forward but slightly down (maybe 10° from the horizontal). Set the color to a pastel shade of your wall color (?, 10, 100). Adjust the brightness to 25 and make this a parallel light. Hide the other lights and render.

This will be a pretty subtle contribution to your scene. However, you will notice how the fill gives some perspective to the scene (in addition to the edge highlights on the vase and shade) and how the more distant floor to the right gets darker. Making the light parallel and positioning it in front of the wall ensures the wall itself is not affected.

Unhide all lights. Activate soft shadows for the key light and render.

Your rendering may look like the one at the right, where not all parts of the scene seem to be casting shadows (check the legs). Here is what you should do.

For the key light, activate the Shadow tab in the Attribute Manager. Change the value of *Parallel Width* to 2000. Render.

Inspect the scene closely. Is the vase casting a shadow on your lamp? If You are happy with the key light placement, move the vase or the lamp instead to fix this. Or, you may wish to do some final positioning of your lights. Do not get too fussy with the illumination right now - we will want to tweak some values after lighting our lamp.

Before moving on to the practical lighting, let's clean up the Object Manager.

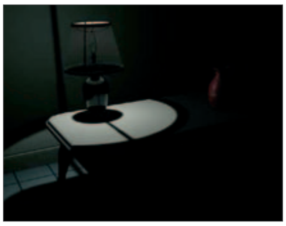
Create a new Null Object and name it scene. Drag all lights and the camera into scene.

Keep a clean scene, stay organized. Good organization is important for efficient workflow.

6. Practical lighting

Here is where we will finally turn our lamp on.

Create a new omni light named lamp shadow and drag it into Lamp. While it is still selected, choose Tools / Arrange Objects / Transfer from the main menu. From the Object Manager, drag the object Bulb into the Transfer To box in the Attribute Manager.



This will move the light to the location of the bulb object. If you placed the bulb's axes in the center when you constructed it, your light will be where you want it. Chances are, your bulb's coordinate system is at the base of the lamp.

Move your light up along the Y-axis to place it at the proper position within the bulb. Turn off the other lights by hiding scene. Render.

This light will need to cast shadows if we want to see realistic effects from the lamp shade.

Make the color of the light about 50, 15, 100. Activate Soft shadows and render.

We gave the lamp a slightly warm color characteristic of incandescent bulbs. Our light needs some falloff so the more distant floor is not as brightly illuminated as the closer tabletop.

In the Details Tab of the light editor, change the Falloff to Linear and set Radius/Decay to 800. Back in the General tab, set the brightness to 150. Render.

If you have studied physics you know this type of falloff should be inverse square. However, linear will allow us to still see some of the lamp's illumination on the floor without adding too much glare to the tabletop. We increased the *Radius/Decay* a bit to compensate for the falloff. You may need to adjust the brightness after activating the other lights and rendering the table's actual texture. You may notice that the lamp shade's wire support casts a rather unnatural shadow. This is because a real light bulb casts light from a volume as opposed to a point, resulting in light coming around both sides of the wire to fill in any potential shadows. Here is how to get rid of the shadows.

Right-click on shade support to bring up a context menu and select CINEMA 4D Tags / Compositing. In the Tag tab of the Attribute Manager, uncheck Cast Shadows. Drag a copy of the tag onto the bulb to prevent any artifacts caused by the light shining through the transparent bulb. You may wish to do this for the lamp shade yoke

as well. (these steps were not done until the end of the sequence of images remaining in this tutorial).

We need the light to shine outward from the shade as well as through the bottom and top openings in the shade. If we made the shade more transparent to allow more light through, we would loose control over its texture and might also see too much of the interior details. So we will add another light to simulate illumination from the shade.

Control+drag a copy of lamp shadow and rename it "lamp illum". Change the shadow type to Soft, adjust the Brightness to 100. Set the Radius/Decay (in the Details Tab) to about 650 and change the Falloff type to Inverse Square.

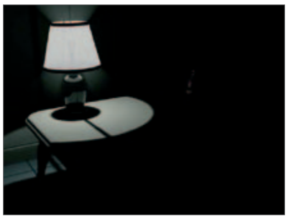
Now we will prevent the shade from stopping the light so the light will appear to come out of the shade. We could do this by not using shadows but the light would also shine through the table onto the floor. We could work around this drawback by making the light fall off rapidly but we would be compromising our ability to exert maximum control over the light's illumination. Let's make use of an excellent feature in CINEMA 4D - the ability to exclude a light's effects from certain objects.

In the Project tab of the Attribute Manager, with the new light selected, make sure Mode is set to Exclude. Click on the shade object in the Object Manager and drag it into the Objects box in the Attribute Manager. Render with both lamp lights on.

Do release the mouse button until the shade is in the *Objects* box. Otherwise the shade will become the active object and replace the light in the Attribute Manager. You may notice in your render that the shade appears too transparent and is not illuminated as it should be.

For the lamp shade material, reduce the transparency to 5. Copy the texture and paste it into the Luminance channel and set the Brightness to about 70.

One last touch for our lamp is to add a bit of visible light. This can give us the look of camera film being slightly overexposed around the brightest part of the light - the effects of a slightly dusty room, or we can use it just to be creative. We will give both lights some visibility. First, let's create an overall lamp glow.



Hide lamp shadow. For "lamp illum", activate Visible in the Visible Light box. In the Visibility tab, set the Outer Distance to 200 and leave Use Falloff checked along with the default value of 100. Render.

This means the visible light will fall off to a value of zero (100%) after a distance of 200 units. This will give us the glow around the lamp. Now, for the final touch, let's send some "creative" rays out of the top and bottom of the shade. This requires a volumetric light with shadows, so we will use lamp shadow.

Hide lamp illum and unhide lamp shadow. Set the Visible Light type to Volumetric. Render.

OK, that looks good. However, it is also a bit intense.

In the Visibility tab, adjust the Outer Distance to about 180 and the Brightness to 75. Render.

You should now recognize the difference between visible and volumetric lights. Let's see what our lamp lights look like.

Unhide the other lamp light and render.

Before tweaking these any more, let's check the lamp light out in context with the rest of the lighting.

Unhide all lights and render.

It is a bit washed out (figure at the right). Both the glow from the lamp and the overall lighting are too bright.

Reduce the Brightness of the visible light (in the Visibility tab) of lamp illum to about 35.

This will reduce the "glow" around the shade.

Reduce the Brightness of the light (general tab) of lamp shadow to 100. Render.

This should have cut down on the glare on the table. Let's put the texture back in the table before deciding we are done.

Delete the temporary Texture Tags you placed on each of the table components. Render.

The overall brightness (figure to right) seems a tad high.

Cut Key's brightness to 90, that of Fill 1 to 30 and that of Fill 2 down to 20. Render

This is good for sun coming in through thin curtains, but if you want to intensify the effect a little, here is how.

In the key light's Shadow tab, reduce the Sample Radius to 1 and render (bottom image to right) .

If you have time, change the key light's Shadow setting to Hard Shadows and render to see the difference. If you really are ahead, try Area Shadows as well. Inspect your scene closely to balance the different lights. Make some changes if you feel it is appropriate. Several lights working together, each with many parameters, makes for a complex situation that may require much fine tuning. Try lights by themselves to see their specific effect but also check what the light does in combinations with other lights.

There are many other approaches you could have taken in lighting this scene. What if you wanted to turn this into a night scene? If we wanted the lamp to be more dominant, as it would be later in the evening, it would need have a stronger illumination. The lamp light is now your key and you set up a couple of appropriate fills. Maybe you would move fill 1 to the right, since the key is now at the left, to simulate the lamp's reflection off walls and the ceiling. You would probably want to change the color of the fills to warmer tones. You may want to add a ceiling light that would behave like a strong fill light overhead. If you have time left, try a night version of your scene (save as "Table Scene Night"). We will return to this scene in the rendering unit. If you are waiting for others to catch up, model some more objects for your scene.

Transparency Tutorial

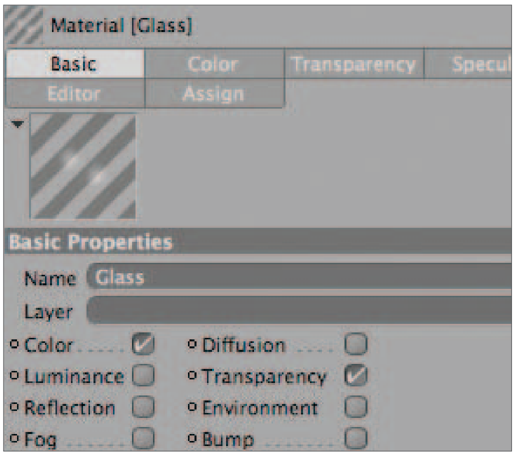
In this tutorial you will learn how to create a typical transparent glass material using some of the advanced transparency functions introduced in CINEMA 4D R11.

Locate the Transparency start.c4d file in Curriculum/Student/ Transparency Tutorial folder.

This file contains a version of the pitcher from an earlier modeling tutorial, a primitive cube placed behind it so that it is partly obscured by the Pitcher. A number of scene elements including a camera and a light object are also contained in the file.

Use the keyboard shortcut Cmd+R to render the contents of the Editor window.

You will see that the light casts hard Ray Traced shadows and the Pitcher object has no material assigned to it giving it an opaque grey appearance. Take a look inside the Scene Elements group to examine the other elements used to create the scene. Next you will create a transparent material for the pitcher.



Material [Glass]

Basic Properties

Name Glass

Layer

○ Color ☑ ○ Diffusion □
○ Luminance □ ○ Transparency ☑
○ Reflection □ ○ Environment □
○ Fog □ ○ Bump □

Create a new Material by clicking in the Material Manager and using the keyboard shortcut Cmd+N. In the Attribute Manager select the Basic tab and type Glass into the Name field and activate Transparency.

You will see that the preview of the material has almost disappeared with just the specular highlight visible. This is because the transparency is set to 100% by default. You will now change this behavior.

In the Attributes Manager select the Transparency tab and set Brightness to 98%. Drag the Glass material from the Material Manager and drop it on the word Pitcher in the Object Manger to apply it. Render the Editor window to see the result.

You can just distinguish the Pitcher as a relatively defined grey shape casting a faint shadow and the cube behind it is now clearly visible. However, the Pitcher looks more like a ghost than a real glass object. This is because we have not defined the refractive behavior of the Glass material.

Click on the Glass material in the Material manager to select it. In the Attributes Manager set the Refraction value to 1.3. Render to see the results.

Now the Glass material is looking more like the real thing. The cube, although still visible, has had its shape distorted by the refractive properties of the Glass material obscuring it and the top lip and base of the Pitcher are now much more clearly defined by dark lines caused by another element of refraction referred to as internal reflections.
If you look in the Attributes Manager you will see that this option has been automatically enabled when the Transparency channel was activated.

However, the shadow cast by the Pitcher seems flat and too even. When light travels through glass it is attenuated slightly by chemical imperfections within the volume of the material and this effect increases at thicker areas of an object. This can create colors within the glass and areas such as the handle should cast subtly darker shadows than the thinner main body.

This behavior can be simulated in CINEMA 4D using the Absorption Color and Absorption Distance settings.

In the Attributes Manager set the Absorption Color RGB values to R = 215, G = 255, B = 195. Render to see the results.

You have now added a slight green tinge to the appearance of the Pitcher and the effect is much more noticeable at the Handle. Looking at the shadow cast you can see that the shadow cast by the handle is correspondingly darker and has taken on the green color from the Absorption effect. Try reducing the Absorption Distance and see how this can change the appearance of the glass material and the shadow cast.

Introduction to Animation

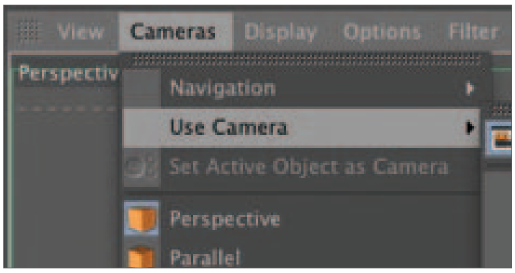
Overview

So far you have been introduced to the basic concepts and techniques that can be used to compose a still image of textured and shaded objects. As well as producing textured, lighted and shaded surfaces from a given set of points, computers are equally suited for moving these points around. Animation adds the fourth dimension (the 4 in CINEMA 4D) of time to your images. This can be used to add life to your scene, tell a story, attract attention or simply to entertain.

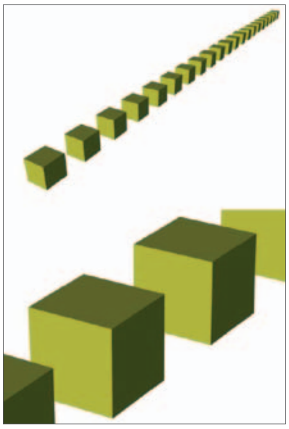
Since an animated scene is always viewed through a camera that can itself be animated, we will begin our look at animation with a brief introduction to the camera object. We will then follow with a look at the most basic and common animation technique: **keyframing**. Keyframing is where an object is assigned motion by specifying its position and rotation at various points in time by setting keyframes.

Although much of the work and most of the visual feedback of animation take place right in the editor window, the tweaking, organizing and orchestrating of complex effects are done in a special animation editor called the **Timeline**. The Timeline is the control center for moving and changing keyframes, sequences and tracks - the fundamental tools of animation. The **FCurve Manager** Works closely with the Timeline and lets you adjust motion by manipulating curves over space and time using graphs. Various parameters of lights, textures, deformers and objects can also be animated in the Attribute Manager as well as in the Timeline and with the **Animation Toolbar**.

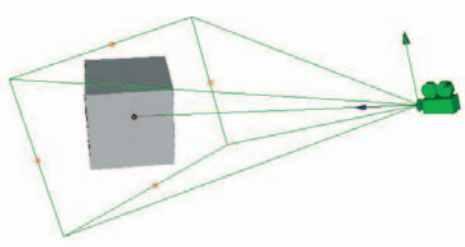
Cameras



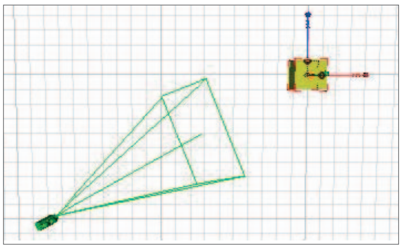
Scenes are always viewed through cameras. Each scene has a Default **Camera** set to whichever angle of view you have chosen - top, front, perspective, etc. However, if you want to animate the camera, protect its position or perform a number of special effects, you must use a **Scene Camera** (or simply, camera). Like any other object, a camera's position and rotation can be animated. Additional parameters specific to a camera such as focal length may also be animated. Even if you are producing a still or an animated scene with a fixed camera it is a good idea to use a camera object. Then you may, at any time, return to the editor camera to view your scene from another angle for positioning or further modeling without disturbing the orientation of the camera through which you will eventually render your scene. You can create any number of scene cameras which can be selected to be the viewing camera from a list under Use Camera in the Camera menu.



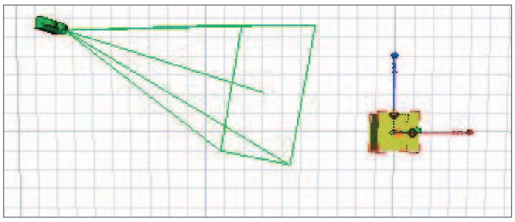
Probably the most important camera-specific parameter is its **Focal Length**. The focal length of a traditional camera lens, as well as one in computer animation, results in a certain angle of view. To the right are shown a wide angle (top) view and a telephoto view of the same scene. Notice that the telephoto view, since it looks at the scene through a narrower angle, expands the center portion of the scene to fill the view, making the objects appear closer. The angle of view, and therefore the focal length, also determines the perspective relationship of objects in your scene. This is addressed in more detail in a separate paper.



The camera views the scene along its positive Z-axis. While in the editor, you may interactively set the camera's direction by moving a handle along the positive Z-axis. You may also interactively adjust the focus (focal length) using the four points on the green rectangle that is projected toward the front of the camera.



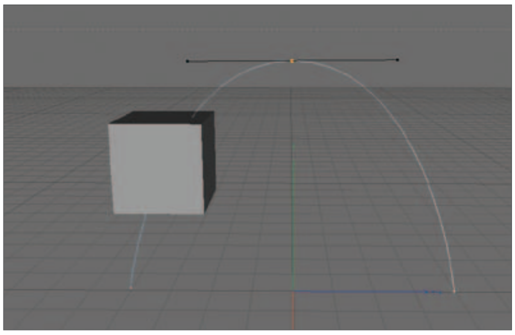
Cameras may be moved and rotated like any other object. However, they may also be moved and rotated using the Camera mode (or keyboard shortcuts). This can result in different effects, depending on which Navigation option is selected is selected under the Camera menu. For example if Object Mode is selected and the camera itself is selected, rotation will occur around the camera's axis center (called a "pan" in cinematography). However, if another object is selected, the entire camera, when rotated with the *3* key (or Camera mode), will swing around the selected object so that the object remains at the same location in the view (three figures to right).



Cameras can also be used for help in modeling or animation. Any number of cameras may be created for use in any viewport. When a camera is selected for the viewport being used as the render view, that camera's viewfinder icon is colored white in the center column of the Object Manager.

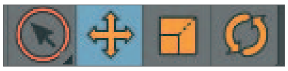
Don't forget that most of the animation concepts and techniques discussed also apply to cameras.

Keyframing

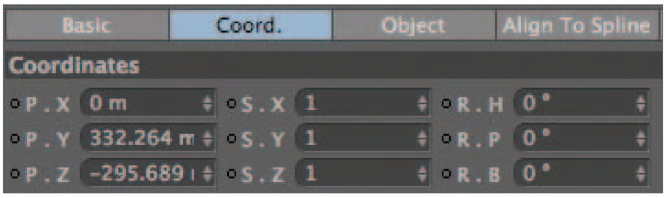


Keyframes, or keys, are the bread and butter of animation. They define movement in the same way points define object surfaces. The positions of objects is defined at selected frames, **keyframes**, and CINEMA 4D computes the positions for the in-between frames. This is similar to what you have seen with Splines - you set the position of the points in space and CINEMA 4D calculates a line through them. With keyframing, the points (keyframes) are positioned over time as well as space. Also, as with Splines, you have the ability to tweak the interpolated motion between points. Hard interpolation results in a straight line movement from one keyframe position to the next. Soft interpolation calculates a smooth curve through the key points. Like other Bézier Splines, you are able to adjust the curve by moving points and adjusting tangents (note the tangent at the top of the cube's path shown at the right).

Keyframes are easy to place and you can use as many as you wish for each object's motion. However, it is best to use as few as possible. Fewer keyframes will, in general, result in smoother motion and allow you to more easily modify the motion while maintaining smoothness.



Creating keys is simple. Select and place an object; choose which parameters to animate in the Animation Toolbar (upper right) and click on the red Record Button. Many other manipulations may be made with keyframes, using the Timeline. Keyframes can also be defined in the Attribute Manager with a Ctrl+click on the circles to the left of parameter names (lower right).

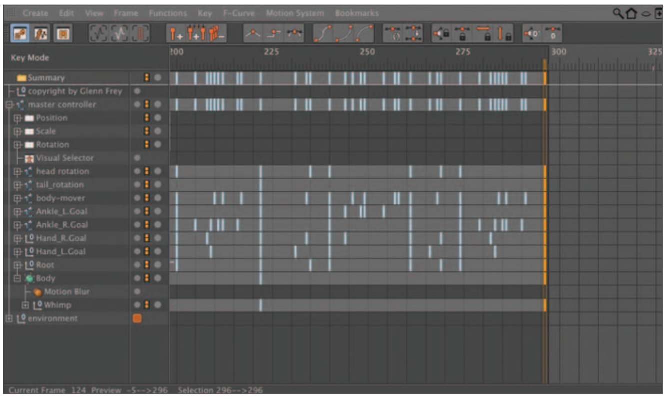


Basic | Coord. | Object | Align To Spline

Coordinates

○ P . X 0 m	○ S . X 1	○ R . H 0 °
○ P . Y 332.264 m	○ S . Y 1	○ R . P 0 °
○ P . Z -295.689	○ S . Z 1	○ R . B 0 °

The Timeline



Current Frame 124 Preview -5-->296 Selection 296-->296

The **Timeline** shows the progress of our animation. In it, each object has its own horizontal track(s) that extend to the right, on that are positioned keys defining its motion. Above the tracks is a scale measuring time in frames. The time units can also be measured in seconds, which is more appropriate for longer productions.

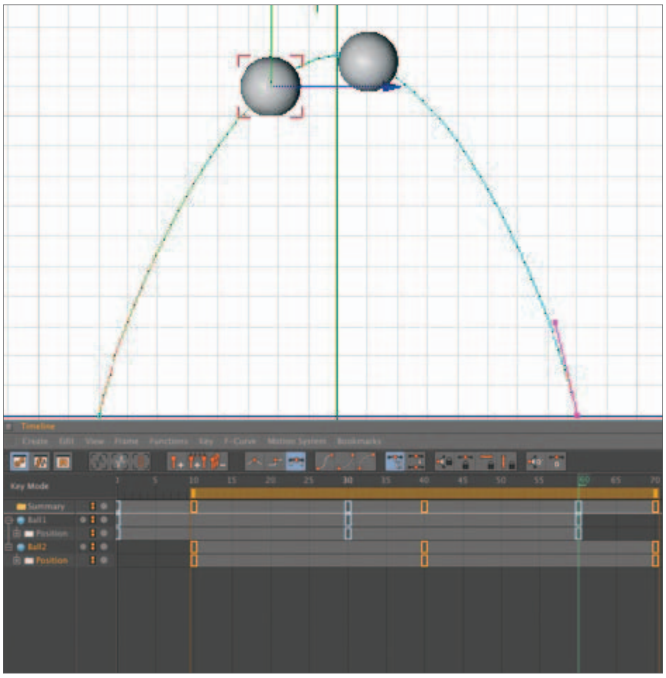
A glance at the Timeline reveals a menu almost as substantial as the main menu. This is sign of the importance of animation in an animation program and of the powerful features available in CINEMA 4D. Many of these are redundant versions of icons in the Animation Toolbar or tools you may never need to use. Others are indispensable in creating and manipulating the basic three components of animation and the Timeline.

Tracks

The term "Track" refers to a particular parameter being animated for a specific object. For example, the object "BeachBall" can have a Position Track with information defining its position over the course of an animation. It could also have a rotation track and, although it probably would not make sense for most beach balls, a scale track. Every animated parameter will have a track. This includes animating a change in texture, changing the visibility of an object, dimming a light or animating a Bend Deformer. Animation tracks particular to each object are hidden by default, but can be seen by clicking the "+" icon to the left of the objects name in the Timeline.

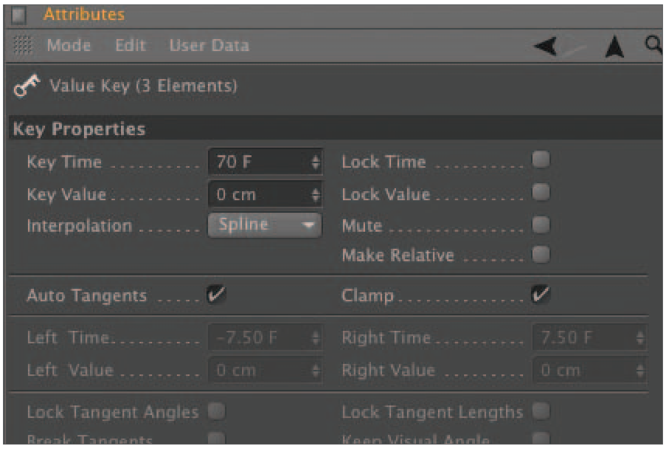
Keyframes

Keyframes (also called keys) are placed onto tracks and define the values for a specific parameter at that point in the animation. Keys can be moved or copied within a Track and copied between tracks of different objects if they are keys for the same parameter, such as position. If you wanted Ball 2 to be at the same position in frame 50 as Ball 1 at frame 30, copy the key in the Position Track of Ball 1 at frame 30 to the Position Track of Ball 2 at frame 50.



Entire Tracks may also be moved and copied to other objects. When keys are moved within a Track the timing of the animation for that object will be shifted. If copied to another object, the second object will inherit the motion of the first. In the example pictured above, the two balls could be made to bounce across the view with the same motion, one following the other If you shifted the copied Ball 1's Tracks to Ball 2 and moved the keys over a few frames.

Entire tracks may also be copied between objects. The keys within these tracks can then be shifted, scaled or tweaked over the course of the animation to give the second object a somewhat different motion from the first.



Attributes

Mode Edit User Data

Value Key (3 Elements)

Key Properties

Key Time	70 F	Lock Time	□
Key Value	0 cm	Lock Value	□
Interpolation	Spline	Mute	□
		Make Relative	□
Auto Tangents	✓	Clamp	✓
Left Time	-7.50 F	Right Time	7.50 F
Left Value	0 cm	Right Value	0 cm
Lock Tangent Angles	□	Lock Tangent Lengths	□
Break Tangents	□	Keep Visual Angle	□

Keyframe markers can also be clicked to load their attributes in to Attribute Manager where their parameters can be modified (right). This would include position and tangent values for a Position Track key. The Timeline also offers layers for organization. These color-coded layers can be assigned to keys or entire tracks. These allow you to selectively show or hide various portions of the Timeline objects as needed to keep an orderly workspace when animating complex projects.

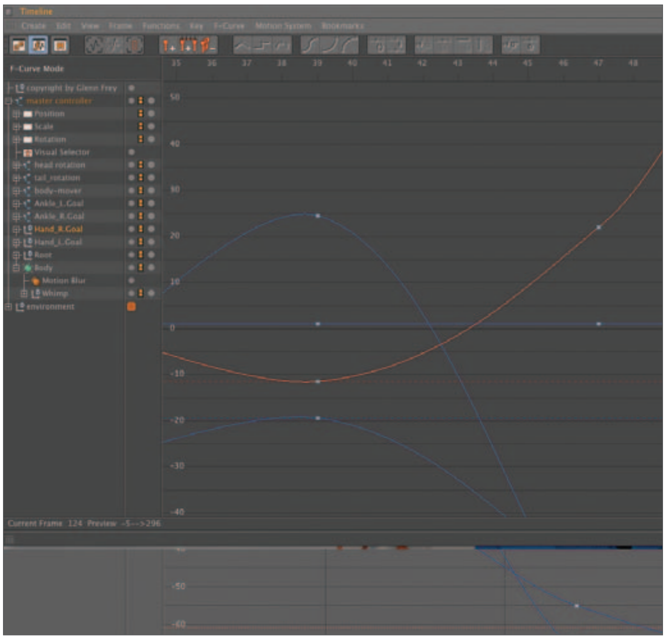
The green Timeslider in the ruler at the top lets you move the view forward or back in the animation. You can adjust the scale and range of the tracks using the *Cmd* and *1*-keys, just as in a viewport. Other features of the Timeline let you edit sequences and view and edit motion curves for certain parameter Tracks.

FCurves

Although you will not find immediate use for these, FCurves is an important and powerful tool. An FCurve lets you view the curve interpolated through your keyframes. More importantly, you can adjust the values of the points visually including the tangents which determine the motion between the keyframes.

CINEMA 4D does not always give you the interpolation you need or expect from a given situation. For example, when an object is meant to be at rest a few frames before moving again, it may reverse its motion a bit before proceeding forward. This is not a bug but a result of equations trying to satisfy constraints such as Soft Interpolation and attempting to produce smooth accelerations. Using an FCurve, you can see these movements visually and make precise corrections.

FCurves allow you the adjust the rate of motion as well. In FCurves, your keys appear as adjustable points with tangent handles (depending on the interpolation you set). Making a curve flatter near a certain key by adjusting the tangents will result in less change of motion at that point. Of course the motion will then be sped up between other frames in order to still arrive at the next keyframe on time.



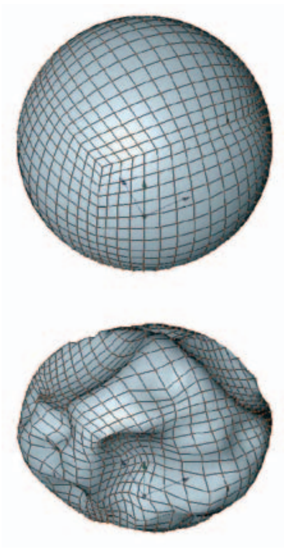
Animating special effects and parameters

Almost any parameter that can be adjusted in the Attribute Manager can be animated creating a **Parameter Track** in the Timeline. The keys on this track will contain values you will typically set in the Attribute Manager. For example, you can adjust the brightness or color of a light, change the angle of the illuminating cone of a spotlight, cause a visible light to extend outward or animate any of the dozens of other parameters associated with lights. The light may, of course, have other standard tracks such as position and rotation. Material parameters may also be animated in the Material Editor.

With a camera parameter track you can change the focus or the depth of field. In a deformer parameter track you can animate the amount of bend, twist or how exploded an object is. Changing the **Growth** parameter of a SweepNURBS Object will cause the object to appear to grow along its path. You could use this to have a name be written across the screen. Need an object to slowly begin to glow? Almost all the parameters of materials and Texture Tags can also be animated.

Morphing and PLA

The above methods move, rotate, scale and change parameters of objects. If you wish to actually change the shape of the object itself, you use some additional tools. Morphing is the technique of blending one object from one shape into another. This must be done using polygonal objects or individual Splines. It cannot be used on primitives or objects containing multiple Splines, such as Loft Objects (unless each Spline in the Loft Object is morphed).

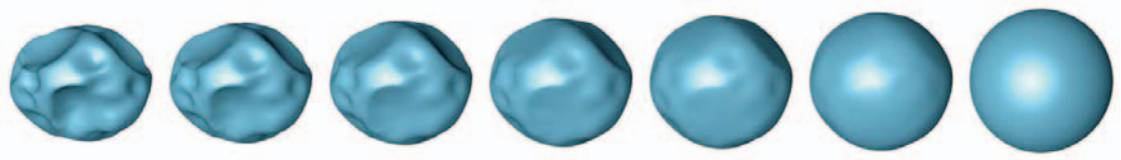


Point level animation (PLA) is also commonly used. This method begins with a PLA keyframe being set for the points at their initial positions. At a new frame, the points are moved to new positions and a new key is set. The movement will be interpolated through all the keys you have created for the object. This gives you precise control over changes in the shape of your object.

Although these methods are valuable tools with many uses, they both suffer from one major limitation: Once you have created your morph, or set keys for the PLA, you cannot easily refine other areas of your object that you want to maintain constant. Suppose you have worked hard to make a frowning face wink. If you then decide the face should be smiling during the wink, you would need to try to model your smile for each keyframe and have them match. Fortunately, CINEMA 4D offers other ways of creating this sort of deformation. There are deformers called "Joints" that can be assigned to various regions of an object's geometry. When the Joint is animated, the animated region of the object is deformed accordingly. CINEMA 4D also offers Spline deformers which use the deformation of a Spline to deform the surface of an object. CINEMA 4D Studio Edition offers other tools, such as the PoseMorph Tag, that lets you mix different deformations simultaneously.

This section has just scratched the surface of animation. As you progress in your learning, you will encounter several other animation tools. You will also see the importance of well-designed objects and hierarchies, as they relate to the animation process.

The following will offer an introduction to the basic process of animating with keyframing. You will also learn some features of the Timeline and Animation Toolbar.



Animation Intro Tutorial

1. Cameras

In this section you will create an Object Camera and investigate various ways of moving and rotating cameras.

Open and adjust CINEMA 4D. Save in the "Intro Tutorials" folder as "Anim Intro". Make sure the perspective viewport is full view and set the editor camera focus to 50.

This will produce an angle of view equivalent to that of the human eye.

Create a Cube Primitive and name it Cube 1. Duplicate it and rename to Cube 2. Position Cube 2 at 400 on the Z-axis. See how quickly you can color the Cube 1 red and Cube 2 blue. Create a Camera Object naming it Cam 1. Make sure the camera is selected in the Object Manager to load its attributes into the Attribute Manager.

You should notice that the camera's *Focal Length* is 50, the same as the editor camera you were looking through when Cam 1 was created. You will also see that you are looking at the view through a green rectangle.

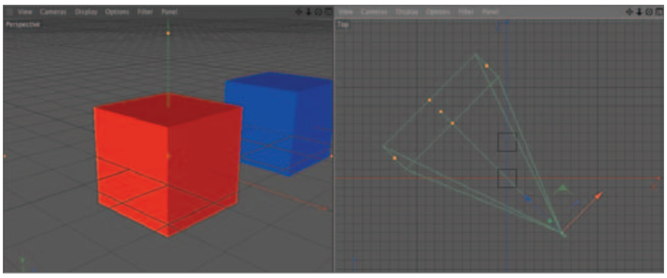
Back the editor camera away by dragging on the editor window with the 2 key pressed.

Now you clearly see the rectangle through which Cam 1 will view the scene. The proportions of this rectangle, the camera's view, are defined by the proportions set by *Resolution* in the render settings. At the center of each edge is an orange dot. These are the camera object's interactive handles.

View the scene through Cam 1 by choosing, from the Camera menu, Use Camera / Cam1.

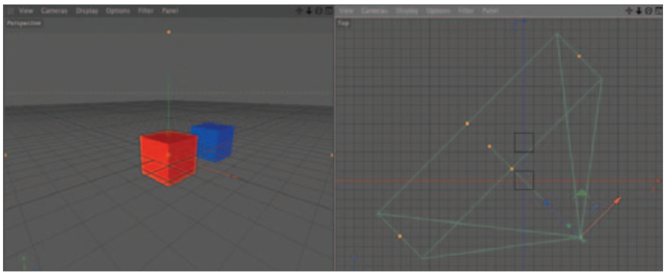
The view should snap back to its state when you created the camera. New cameras are always created using the point of view and focal length of the active viewport. In our case, this was the editor camera whose focus we set to 50.

From the view menu choose View / Panels / 2 Views Side by Side. Move out in the top view (right) until you can see all the camera handles.



The left viewport should be the perspective view from your Object Camera (cam 1). The right panel should be a top view from an editor camera. In the top view you can see the camera and its green interactive handles.

Grab any one of the four points on the green rectangle and move out away from the center of the rectangle.



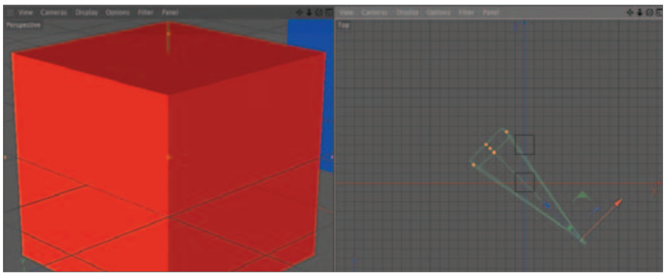
You should instantly notice the camera-eye view on the left changing to accommodate the change in focus. Because the angle of the view is larger the objects take up less of the viewing area and so appear smaller as can be seen from the top view. Lenses that view a wide angle are called wide-angle lenses and are said to have a wider field of view that refers to how much of your scene can be seen through the camera.

Inspect the focal length in the Attribute Manager.

The focus of the camera should have changed to a value less than 50. This is not the same as moving the camera in and out using *Cmd+1+drag*.

Back in the top view, move one of the frame handles toward the center of the rectangle until you have a narrow viewing angle.

The object now appears much larger because it occupies most of the narrowed viewing angle (taking up most of the field of view). This type of lens is called a telephoto lens (but not a "narrow angle lens").


View Cameras Display Options Filter Panel
Perspective
View Cameras Display Options Filter Panel
Top

Reset the Focal Length to 50.

When the viewport through which the camera is viewing the scene is active, and the Camera object is selected, you can also set the focus in the Coordinates Manager.

With Cam 1 still selected in the Object Manager, choose the Model mode and the Move Tool. Translate the camera back and forth along its Z-axis in the top view.

In the camera-eye view to the left, your cubes should get larger or smaller as the camera moves toward or away from them. If you are perceptive, you will notice that this has a different effect than changing focus (zooming). Zooming, in addition to making the cubes appear nearer or farther, changes the cube proportions to one another. A telephoto lens makes more distant objects appear closer to those in the foreground in both distance and size. A wide-angle lens, on the other hand, will exaggerate relative size and distance.

Undo. Choose the Rotation tool with only the heading axis unlocked. Make sure you are still using the Model mode select the World Coordinate system and make sure Cam 1 is still selected. In the top view, rotate the camera object. Do the same in the perspective view.

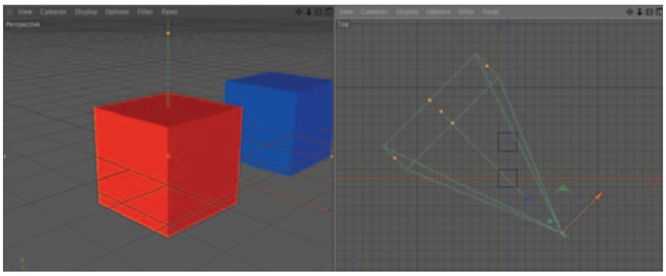
This produces different results because when in the top view manipulating the Y axis alters heading, however in the perspective view the HPB system is used so that the Y axis now refers to the Camera's bank parameter. Except for the camera's interactive handles, all your manipulations of the camera so far have been like that of any other object. Now technique becomes a bit more specialized to cameras.

Undo all rotations (hPB = 45, +20, 0). While holding down the 3 key, move the mouse in the perspective to rotate the view.

The Camera now has complete freedome of movement ignoring the fact that we disabled the X and Z axes. Since the view is through Cam 1, the camera is rotated and moved as a result of rotating or moving the view (using the 1, 2, and 3 keyboard shortcuts is the same as using the Camera mode).

Now try an undo (*Cmd+Z*).

Nothing happened, at least the most recent rotation was not undone. Although you are still moving the camera, CINEMA 4D still treats this as a view move for undo purposes.

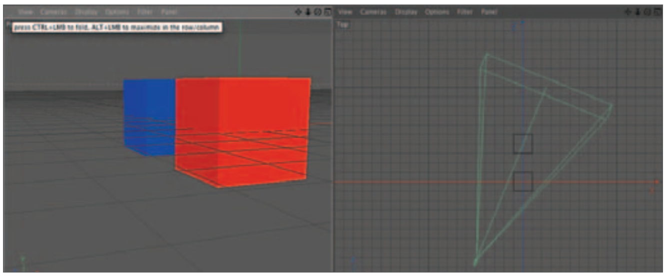

View Cameras Display Options Filter Panel
Perspective
View Cameras Display Options Filter Panel
Top

Undo using Shift+Cmd+Z.

The reason camera movements are treated differently from movements of the view is:

In the Perspective view (which is still viewing through Cam 1), rotate the view via the 3 key and a click and drag anywhere on Cube 1.

Look at the top view after rotating in the perspective view. The camera is not only rotating, it is translating side-to-side to make sure Cube 1 stays in the same relative position in the camera's view. Imagine trying to do this by moving and rotating the camera object in the top view. This is not only useful in animation but very helpful in the modeling process as well.


View Cameras Display Options Filter Panel
press CTRL+LMB to fold, ALT+LMB to maximize in the row/column
View Cameras Display Options Filter Panel
Top

Undo. Click on Cube 2 and rotate again using the 3 key. Undo.

A point selected on any surface using the cursor becomes the point of rotation for the camera. If there is no geometry beneath the cursor then a point is chosen in the centre of the editor window at the same distance from the camera as the previous selected point. Translation and scale changes, when done within the camera's view, must also be undone using *Shift+Cmd+Z* (*shift+Cmd+Z* works on the currently active view).

If you are setting up a big camera move or you want to lock a particular axis of the camera's movement, you may want to use the Camera mode. Important reminder: Do not forget the shortcuts.

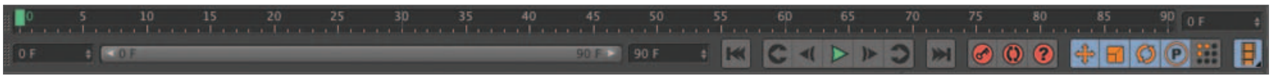
Cameras/Views: Move-Scale-Rotate. It is as simple as "1+2+3".

There are similar shortcuts for object manipulation but they are not quite as versatile as the tools in the toolbar.

Objects: Move-Scale-Rotate. It is as simple as "4-5-6".

2. The Animation Toolbar and keyframing

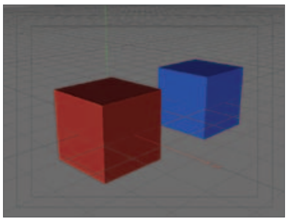
The Animation Toolbar offers two principal functions. First is the ability to record keyframes for a selected object. The solid red spot is the *Record Keyframe* ("record") button.



The second function of the Animation Toolbar is playing or navigating through your animation. This is done with the buttons at the left of the Record Button and the Timeslider ("scrub bar"). Let's begin with a simple animation of one of our cubes.

Move the camera so your view looks like the one to the right.

You should still be using the scene camera you created earlier (cam 1).

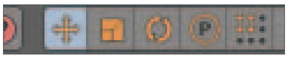


Hide the camera object (cam 1).

The panel view should still be a two-view mode with the top view at the right. Even though Cam 1 is hidden to clean up the top view, you are viewing through it in the perspective viewport on the left.

In the Animation Toolbar, make sure the only box for the animatable parameters (to the right of Record) checked is that for position (the first in the series).

The button at the far right is not a part of the series of recordable tracks. With the following steps, we are going to animate Cube 2 to travel in a circle around Cube 1.



Be sure Cube 2 is selected. With just the Position button active and the Timeslider set to zero, click on the Record Button.

You have now created a Position Track for Cube 2 with a keyframe at frame 0.

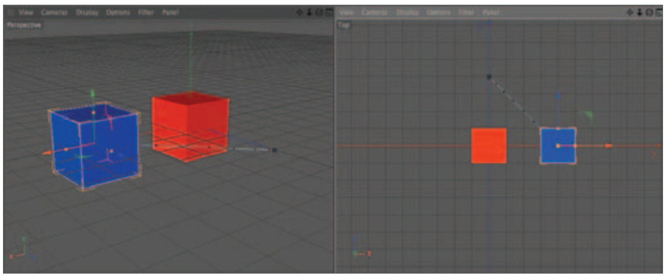
You may easily change this later, if needed. Check the Attribute Manager. You should now see three red dots left of the position (*P*) parameters for the *X*, *Y* and *Z*-axes. These indicate keyframes have been recorded.

Move the Timeslider to frame 10. From the top view, move the selected cube to 400, 0, 0 (X, Y, Z).

You can enter the values directly into the Attribute Manager or Coordinates Manager (make sure to finish with a click on Apply or press Return on your keyboard). This brings Cube 2 to the right of Cube 1. Monitor the cube's position in the Attribute Manager or Coordinates Manager as you set the keyframes.

Move the slider back one frame then move it back to frame 10.

You lost your new position! This is because you did not record it. The animation track currently has the object at the original position, as recorded at frame 0. You need to record a new keyframe to specify a new position.

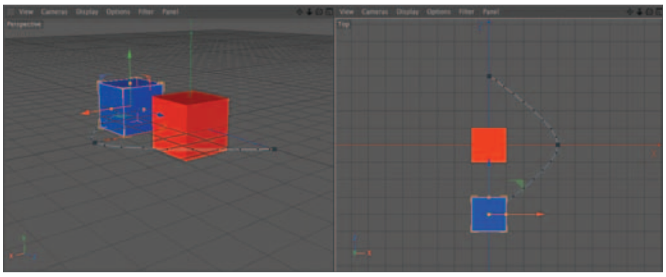

View Cameras Display Options Filter Panel
Perspective
View Cameras Display Options Filter Panel
Top

Move the object again to 400, 0, 0.

This time we will use a different method of recording our keyframe. Notice that the red dots associated with the position coordinates in the Attribute Manager have turned to open circles. This means they have recorded tracks but do not have keyframes at the current time.

While holding down the *Control* key, click in one of the grey circles left of the position input boxes.

A click on any P will highlight all three. When this is done, a keyframe will be added to all three tracks. If you wish to add a key to only one track, click its name (*X*, *Y* or *Z*) to select and add a key to only that axis.

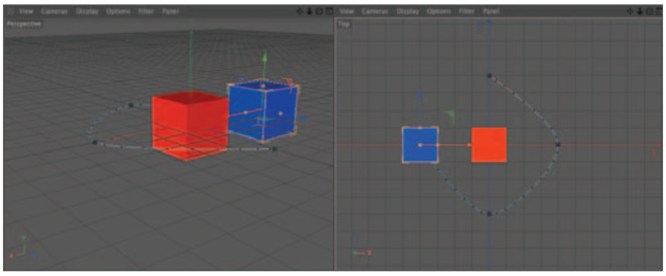


At frame 20, move the cube to 0, 0, -400 and record. Set another key at frame 30 with the cube at –400, 0, 0.

That is all for now with the keyframes. Let’s see what you have done.

Drag the slider back and forth. When done, leave it at frame zero.

Now you are animating! You will also notice the animation path in blue which shows whenever the animated object is selected. The two green buttons in the Animation Toolbar are for playback.



Click at the right pointing, green triangle button to play forward. Stop playback after frame 30.

Hope you figured out the play button changes to a pause button once pressed. I’m guessing you know what the left-pointing triangle does.

Click on the button to the far left. Press Cmd / G on the keyboard.

This is the *Go to Next Key* frame shortcut, it should have taken you to frame 10, your first keyframe. Predict which frame you will go to if you use this shortcut again.

Press Cmd / G again. Cmd / F is the Go to Previous Key Frame shortcut.

When animating, it is easy to swamp your screen with windows (as we will see after opening the Timeline, making it hard to get to the scrub bar). Let’s use the Heads-Up-Display (HUD) to create a moveable frame scrubber.

Press Alt+V to bring up the Configure All choices in the Attributes Manager. In the HUD tab, check Frame.

You should see a frame counter appear in both windows (you can limit this to only the perspective view by using *Configure* instead of *Configure All* under the viewport Options menu). This does more than just display the current frame.

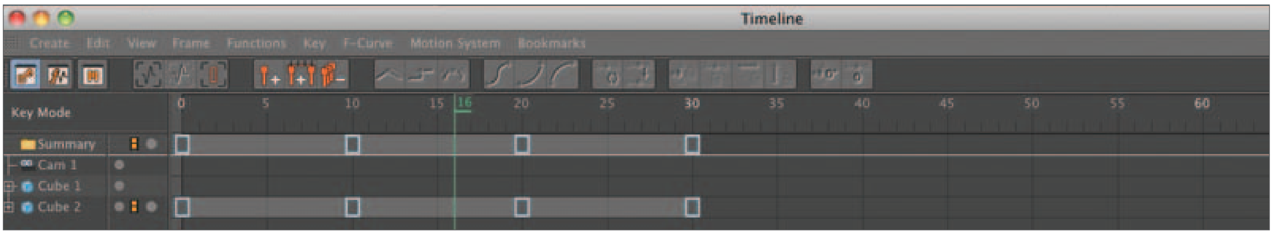
Click and drag right and left within the Frame HUD.

Looks like it may also be hidden by any window which covers the Animation Toolbar.

Ctrl+click and drag the Frame HUD within the view.

In CINEMA 4D you can add a HUD item for almost any parameter. Simply left-click on any parameter in the Attribute Manager and drag it into the editor window to add it to the HUD. Let's take a look at the Timeline to really help visualize the process of keyframing.

3. The Timeline



Introduction

This is the control center for animation. Not only does the Timeline offer access to all the important functions of the animation process, it also offers a visual layout showing how the tracks of the different objects relate to each other. Although you can tab and dock the Timeline like any other manager, for larger projects it takes up a major portion of the screen and so must be opened and closed frequently. For this reason I give it another shortcut in addition to the native *Shift-F3*.

Using Shift+E (non-standard), open the Timeline. Use Shift+W (standard) to close the Timeline. Open the Timeline again.

By default all objects which can be animated, including materials, are shown in the left column. From the Timeline menu, click on the *View* menu and inspect the choices.

Choose Show Animated.

Notice how this cleaned up the clutter in the Timeline. As you continue to animate additional objects, the list may become cluttered once again. You can then select Link to Object Manager to show only objects that are currently selected. The Automatic Mode menu item acts as an on/off switch allowing you to display custom groupings of objects, which is very handy with large projects.

Cube 2's Tracks are nested below its name. If you expand this item you will see that Cube 2 has three items below its name. One is labeled *Position*, which we will refer to as the Master Track. This contains three tracks, one for each component axis of Cube 2. Keyframes ("keys") contain the information pertaining to the location of the object along that axis at that particular frame. If you look above each key in the Timeline ruler you will see that the frame at which each key is placed corresponds to the location at which you positioned the Timeslider while placing keyframes earlier.

With the *1*-key held down, click and drag anywhere in the ruler (avoid the green "time marker") or an empty region in the third column.

This is just like moving the view in the viewport.

Re-size and reposition the Timeline window, if necessary, so you can also see the perspective view and the Animation Toolbar. Grab the time marker (a number next to the green line in the ruler of the Timeline) and drag it forward.

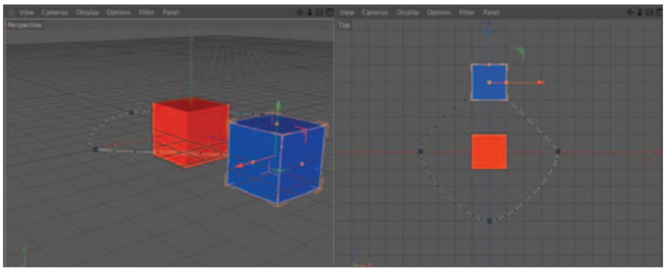
You should see your cube move, as well as movement of the slider in the Animation Toolbar and the value in the HUD change. Before going on, a word about components. This term refers to the individual values which combine to describe a parameter such as position. For example, the position of an object in 3D space is a combination of X, Y and Z position values. This works the same for rotation (combination of H, P, and B components), scale (along the X-axis, Y, and Z) and even color (R,G,B). Some operations in

CINEMA 4D work directly on the combined value but the components can also be controlled individually as described earlier.

Placing keys in the Timeline

You have already created new keys using the *Record* Button in the Animation Toolbar. There are several ways to make a new key in the Timeline. You can Ctrl+*click* on the Track at the location (frame) at which you want the new key inserted. This will create a new key at that frame using the current values (of position in this case) of the object in the view. Let's try this method to complete our circle.

Ctrl+click on the orange key at frame 0 on the Position Master track for Cube 2 and drag along the Master Position Track, releasing at frame 40. Scrub to see the effects.



You can also see the animation path has been extended in the view panel. The position of the object at frame 40 is exactly that of frame 0 which was where the new key was copied from. Ctrl+drag copies keys as well as objects.

Maybe you missed the exact frame 40 when clicking to set a new key.

The newest key should still be highlighted in orange with two orange lines extending vertically. If not, click on the key to select it. Observe the status bar at the bottom of the Timeline. The value for Selection should read 40.

In the Attribute Manager make sure Mode is set to Timeline and set the Key Time to 40.

Now we will look at a method of moving all the keyframes associated with one track.

Click on the Position.Y item in the left hand column to select all the keys in that track. Click on any of the selected keys and drag to the right to move all selected.

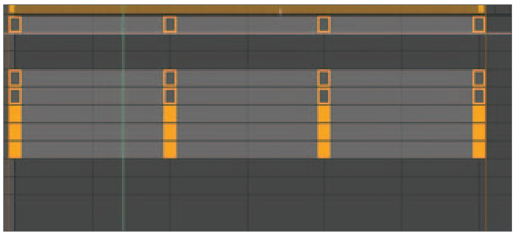
All keyframes contain their previous values but the values will now be achieved at different times.

A look at the animation path shows our circle is not very circular. Although we could have made a perfect circle from the start, let's adjust what we have to learn some new techniques.

Keys and curves

How does CINEMA 4D know what curve to draw between the keyframes? Placing points in-between other defined points is known as interpolation. The interpolated points making up the curve are currently calculated with the default interpolation. You can change this.

Delete the keys at frame 40.



May as well get the curve right at frame 0 before copying to frame 40.

Begin below and to the right of any keys and drag a selection rectangle around all the keys.

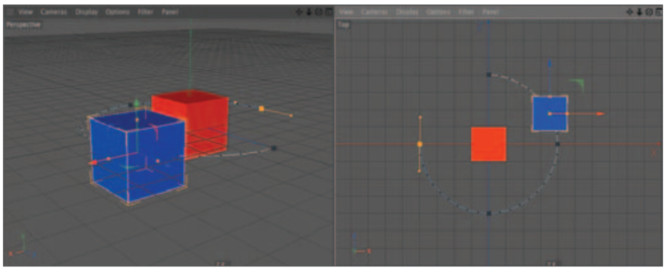
You can add to or remove from this selection by *Shift+clicking* . You will notice the Attribute Manager is now filled with parameters for adjusting the interpolation of the keyframes. The Time and Value boxes should be empty since the keyframes contain different information for these values.

From the *Key* menu at the top of the Timeline choose *Spline Types / Soft*.

This will reveal Spline handles in the editor window, allowing us to interactively edit the animation path of Cube 2.

In the top view, click on a point coinciding with a keyframe.

These are hard to spot. A tiny yellow box on a blue line. Look in the region of where you know the points should be. There will be two handles for each point that affect each side of the point symmetrically.



Drag the yellow handle until your curve looks circular at that point. Do the same for the other three points.

You may want to move the object by moving the slider to better select the keyframe points.

Again, copy the keys at frame 0 to frame 40.

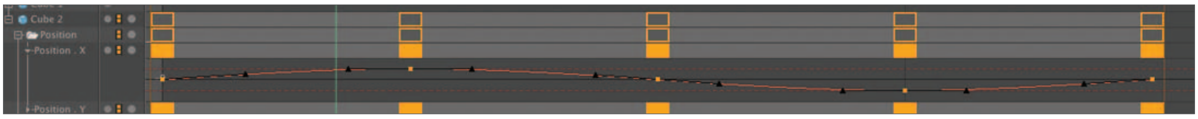
We now have a circle. As your animations get more complex, you may need to learn about other interpolation methods and strategies from the Help files.

Immediately to the left of the Track name in the Timeline is an expand/collapse box.

Expand the *Position.X* track (click on the + just left of the track's name).

This is called a function curve (FCurve) and allows you another way to adjust the interpolation and values.

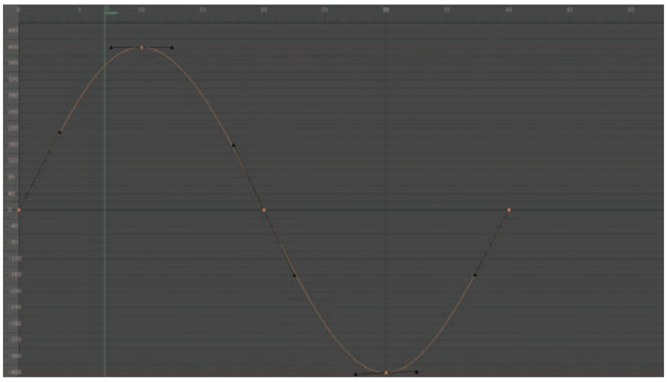
In the FCurve, click and drag the point at frame 10 upward. Observe the changes in the view panel. Undo. Do the same for the point's tangent. Undo.



Well, this could be handy. Too bad it is so crimped.

Make sure the Timeline is the active window and press the space bar. If necessary, select Cube 2 in the Object Manager to load its tracks.

You may also select multiple objects to share the window. Welcome to the FCurve Manager. The "view" in the window may be moved and scaled in the same way as the viewport and Timeline, with combinations of the *1* and *Cmd*-keys. You can also unclutter the display by selecting any combination of individual tracks from the left side list. Click the object's name to display all of its tracks. This is very versatile and powerful. You can move points (keyframes) in space and time, adjust handles, change interpolation (via the Attribute Manager) and more. FCurves give you ultimate control not just of position and rotation, but of how lights dim over time, how a material changes colors and basically anything which is animated and has a value associated with it. Most importantly, FCurves give you a graphic relationship between different tracks and allow visualization of how motion varies over time. For example, in a Position Track, the slope of the curve indicates the velocity. In the figure at the right you can see the object zipping away from frame 0, covering lots of space in a few frames. However, as it approaches frame 10, the slope of the curve flattens, showing it is slowing (of course, it is speeding up along the Z-axis at the same time). You can also see that the curve appears a little sharp at frame 30



compared to the nice roundness at frame 10. We can easily adjust that here.

Play around a bit with points and tangents while viewing changes in the view panel. Undo all changes when done (unless done to improve the circular path). Close the FCurve Manager with Shift+W when done.

Think for a while before proceeding. How could you make the cube continue in a second orbit? Try it before resorting to the fine print below.

Copy the frames from 10 to 40 to occupy frames 50 to 80.

Copying keys and tracks

Selections of keyframes and even entire tracks may also be copied. First, let's set up the Timeline so we can access Cube 1 as well. You could select *Link View to Object Manager* from the Timeline's *View* menu item and select both cubes in the Object Manager. This method will work fine but here is another:

Uncheck View / Automatic Mode from the Timeline menu.

Now we can customize the Timeline display to show only the objects we want, regardless of which become selected in the Object Manager.

Drag Cube 1 **from the Object Manager into the left column of the Timeline. Do the same for** Cube 2**.**

The display will remain like this until we switch back to automatic mode. Drag another object into the Timeline or remove one from the timeline by Right *clicking* on the object's name and using *Remove Selected Objects* from the contextual menu (this feature means very little with only three objects, but when you have hundreds of objects you will be thankful!) Now to make Cube 1 also revolve around the origin.

Select Cube 2's position keys. Ctrl+click and drag the keys up to the line above to the right of Cube 1's name.

There will be no result, which means the operation was not possible. This is because Cube 1 has no Position Tracks, yet. We could create an empty Track using *File / Add Property Tracks / Position / All*. However, since we know we want to fill it with the existing keys, here is a better way:

Ctrl+*click* on Cube 2's Position Track name and drag the entire track straight up to the Cube 1 line. Scrub forward and back.

Both cubes now occupy the same positions at the same time. Let's stagger their motion.

Ensure all the new keys for Cube 1 are selected, click on one and slide the selection to the right so it begins on frame 20. Scrub to see the results.

Things look good after frame 20 but we need to have Cube 1 moving from the very beginning.

Here you need to plan a bit. You know you need a key at frames 0 and 10 and that two of the keys already in the keyframes will work. See if you can put this puzzle together.

Copy the correct two keys to frames 0 and 10.

Did you get solve it? (look at the figure below for a hint). One strategy: Set the view to frame zero, copy what you think is the proper key to frame zero and view the result (you want Cube 1 directly opposite Cube 2 on the other side of the World origin). Once you have the key at frame 0, the second key should be easier. This is also a good place to check the FCurve Manager. There you can see the repetitive nature of the motion.

Delete the Cube 1-keys at frames 50 and 60 to make the animation tidy.

CINEMA 4D also offers methods to loop motion so you can have the cubes continuously revolve using just a few keyframes. If you need this feature, refer to the help files. If you are ahead, see if you can extend the circular motion of both cubes to frame 80 using the Timeline.

4. Animating parameters

We have been using the term "parameter" a bit loosely to include position, rotation and scale of an object. In this section, a parameter is more accurately referred to as those values that are used by internal formulae to create or modify certain objects. This includes values such as the height of a parametric cube, the amount of bend in a Bend Deformer, the brightness of a light and the focal length of a camera.

Animating a deformer

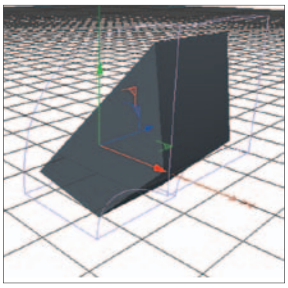
Like position, scale and rotation, parameters can be keyframed using the Attribute Manager. Here is an example using the Bend Deformer.

Use a single perspective view for the following. Hide the two Cube Objects and create a third cube named Cube 3. Apply a Bend Deformer (bend) to the new cube. You should be at frame 0 with the Bend object selected.

We want to see what's happening in the Timeline even though we will use other methods to create the keys.

In the Timeline, check View / Automatic Mode and choose Link View to Object Manager also from the View menu.

Here is your first keyframe in the Attribute Manager.



In the Attribute Manager, Ctrl+click on the Strength parameter's animation dot. You can also right-click to bring up a contextual menu. Choose Animation / Add Keyframe. This is less efficient but also allows access to other functions when you need them. Scrub to frame 30.

Notice the dot is hollow indicating a track but no keyframe at this frame.

Change the Strength value to 90 and record a key. Scrub in the Timeline to view the results.

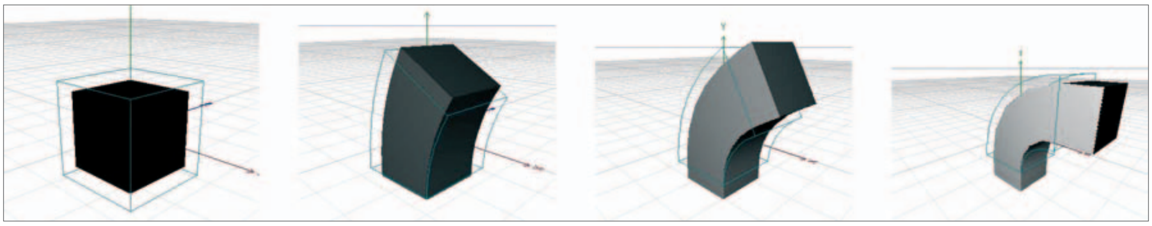
You can still adjust the value interactively in the viewport while keyframing in the Attribute Manager. You are probably wondering why the bend looks so clunky. In this state, it should rather be called "slant". Make it bend by changing a single parameter of the cube.

Just as with a position or rotation track, you can edit points and handles in the Timeline or, better yet, in the FCurve Manager. If you wanted to change the Strength value at frame 30, you would need to change the value, then Ctrl+*click* the animation dot again to register the change. Here is another way to change the value:

In the Timeline, select the key at frame 30 for the bend's Strength.

This loads the keyframe data into the Attribute Manager.

In the Attribute Manager, change the Value to 120. Scrub a bit and return to frame 30.



When manipulating the value of a key directly in this manner you do not have to record in a separate step.

Any parameter displayed in the Attribute Manager that has a small dark circle in front of its name can be animated. For many objects such as the Bend Deformer, this means everything but the name!

Cameras and lights

The position and rotation of cameras and lights are animated in the same way as other objects. Other specific parameters of each, such as focal length of a camera or brightness of a light, can be animated using parameter tracks created in the Attribute Manager.

Animate Cam 1 Focal Length to change from 50 at frame 0 to 25 at frame 30. Play.

This is best done in the Attribute Manager.

Create an omni light and position it to shine from above and left of the camera (at about 400, 300, -600). Animate its brightness to change from 0% to 100% over the first 30 frames.

You should notice tracks and keyframes appearing where expected in the Timeline. Let’s revisit the FCurve Manager.

Ensure the Timeline window is the active window, open the FCurve Manager (space Bar) and unfold the light parameters, click on Intensity to load its keys into the view.

Between frame 0 and frame 30 you should see a curved line indicating that the brightness changes in a non-linear fashion.

In the FCurve Manager, Ctrl+click on the line at frame 10.

A new key should be inserted.

Drag the new key around in the window a bit. Leave it positioned at about frame “15” with a brightness of about 20.

Now the light will brighten slowly at first, then get brighter much more rapidly toward the end of the track. If you need more control over the curve you could either add more points or convert the existing points to a Spline (soft interpolation).

5. Animating materials and textures

A texture can be animated to move, shrink or rotate on the surface of an object using the Texture Tag’s attributes in the Attribute Manager. Most parameters of a material can also be animated in the Material Editor as well as the Attribute Manager (the notable exception to this is that you cannot change texture maps assigned to channels).

Return to frame 0 and hide the light you created above. Create a new material, make its color green and assign it to Cube 3. Double-click on the material in the Material Manager to open the Material Editor.

Materials can be animated in the Attribute Manager but the Material Editor is better optimized for efficient manipulation of materials.

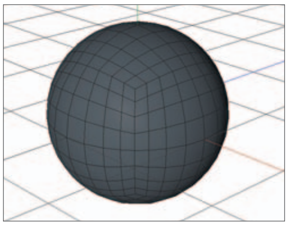
Activate the luminance channel; change the Color to red and the Brightness to 0. Record a keyframe for Brightness. Scrub to frame 30 and set the luminance Brightness to 100 and set a keyframe. Scrub, or play, to view the results.

There are dozens of other parameters to experiment with. Try some if you are ahead of the class. Some of them need to be rendered to see the results, e.g. when changing the bump strength or the amount of glow.

You can even work around the restriction of not being able to animate the texture maps. CINEMA 4D also lets you dissolve between two different materials. A dissolve is where one effect (or scene) gradually fades out at the same time as another fades in to replace it. You can do this by creating a track for the Material option of the Texture Tag, dropping in different materials at different times and keyframing those (you will probably not need this feature anytime this year, so no elaboration here).

Materials themselves may use animated textures maps. The texture maps used in different channels of a material may consist of QuickTime movies or other animated formats which allow the texture to be animated independently of any texture tracks. A common example would be having a movie displayed on a television in a living room setting. Only one material is needed containing a movie, usually in the luminance channel (better than the color channel since TVs emit their own light). This is also how you could use CINEMA 4D to record an animation over a piece of live action footage although this is more typically done with post-processing software such as Adobe's After Effects or Apple's Final Cut Pro.

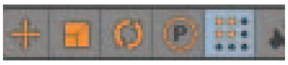
6. PLA (Point Level Animation)



PLA and morphing are used to manipulate the shape of individual objects. PLA directly affects the points of objects. We will just take a brief look at the basic principles here.

Create a sphere, change the Type to Hexahedron and leave the segments at 24. Make the sphere editable and select the Point mode.

smoother surface as we move the points around.



In the Animation Toolbar, make sure the only box selected for recording is the PLA icon. Move to frame 0 and record.

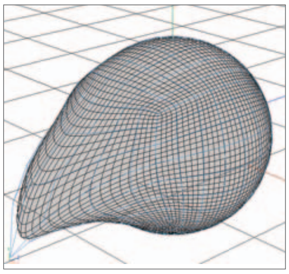
PLA animation can only be done using the Animation Toolbar. All adjusting of points is done in the editor window - there are no parameters accessible in the Attribute Manager.

Scrub to frame 30. From the pop-up menu in the editor, choose the *Magnet*. Pull at the points of the sphere until you achieve a shape which no longer resembles the sphere. For finer adjustments, reduce the *Radius* value for the magnet in the Attribute Manager. Record.

Thirty seconds and you can have almost created the head of a weasel. You must be careful with regard to undos when animating within an existing sequence. A single undo will undo all movements of the object or, in the case of PLA, points, and return the values to those of the existing keyframes. When creating a new key it is safest to record after you have made a change you are satisfied with, even if you want to keep refining the points at that frame.

Drag the sphere into a HyperNURBS Object to smooth it out and play or scrub through the 30 frames.

One disadvantage to PLA is that, once you have created a PLA track, any changes you have made to the points will apply only to that frame. Hence, if you wanted your weasel head to have a bump in the back, you can add that bump at frame 0 but it will not be there at frame 30. More advanced techniques such as bones and Spline Deformers will allow you to adjust various portions of the mesh independently.



Morphing is similar to PLA in its effect. Instead of manipulating the points of one object over time, with PLA keys, morphing lets you store a collection of shape variations within a single *Morph* tag, then mix between the stored variations to achieve a much larger variety of shapes. Check it out in the help files.

There are many other methods and tools used in animation but this survey has covered the basics and will allow you to perform most of the animation you will need to get started.

Sun, Earth, Moon Tutorial

In this tutorial you will practice using Null Objects in animation by creating a model of the sun-earth-moon system and using Null Objects to help simplify the rotations and revolutions involved.

Overview

Revolution occurs when a body revolves in a circular path around another body. A good example of this is the earth's yearly path around the sun - one revolution per year. The moon revolves around the earth approximately every 28 days.

Rotation happens when a body turns around its own center, which is called the "axis of rotation". The earth rotates every 24 hours around its axis, which runs from the north pole to the south pole. The moon also rotates, but we always see the same face. This is because it rotates once per earth rotation. You will see this as we work through the tutorial. From studying features such as sunspots, we know that the sun also rotates once every 25 days or so.

In this project, we will get the rotations and revolutions correct. That is, they will be in proportion to each other, just as in the solar system. However, we will not try to be realistic about the relative sizes of the bodies nor the distances between them. If we were to do so, they would appear only as tiny dots if we wanted to fit them all in the view.

The sun and earth

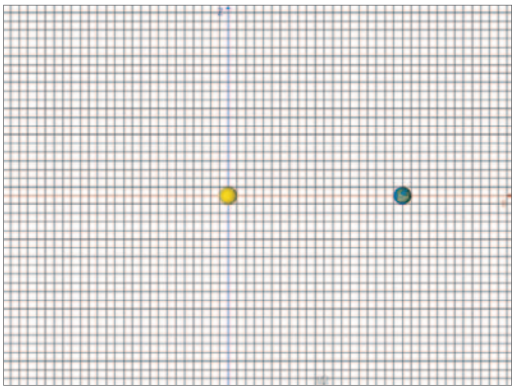
Let's begin by setting up the sun and earth.

Create a new folder named "SunEarthMoon" and, within it, a new "tex" folder. From your Student Folder, drag copies of "Earth.color", "Earth.b&w" and "Moon.tif" into the new "tex" folder.
Now create a new CINEMA 4D file named "SunEarthMoon_1". Set the Focus to 50. Create a primitive sphere named Sun and leave the values at their default. Create a new yellow material named sun and apply it to the sun object.

This object is temporary - you will replace it with lighting effects later.

Create a second sphere named Earth also using the primitive's default values. Make a new material called earth and apply the texture "Earth.color" to the Color channel.

Apply this new material to the Earth object. Move the Earth out along the X-axis a distance of 2000. Switch to the top view (with Gouraud Shading active) and move the camera back. Keep the Sun centered and make the Earth visible.



The year

Now we will set up the revolution of the earth around the sun. If we create a rotation track for the earth right now, the earth will rotate around its own axis. If we move the earth's axis to the World origin where the sun is positioned, then any rotations of the earth object will result in the desired yearly revolution around the sun. However, then we could not also effect the earth's rotation around its own axis to create the day/night cycles. It seems like we need two axes of rotation. Let's leave the earth's axis where it is to use for the daily rotation and create a Null Object to use for the yearly revolution.

Create a Null Object and name it Year. Make Earth a Child of Year. Lock the pitch and bank axes and rotate Year about its heading.

You should be able to revolve the earth around the sun, just as the earth does in its yearly orbit. Now let's set up the animation.

Undo any previous rotations. Open the Timeline and, from the CINEMA 4D main menu, choose *Edit / Project Settings*. Set the *Maximum* value to 3650.

This is an awful lot of frames but we are not going to render them all. We are choosing numbers here to make our calculations easier. There are 365 days in a year. We could animate the earth to go around the sun once in 365 frames. However, to then show how the earth's *rotation* results in the day/night cycle, the earth would rotate once every frame. This means it would be back in approximately the same rotational position each frame and would not appear to rotate correctly. This is the same phenomenon as seen in motion pictures when helicopter blades appear to move slowly or not at all or when wagon wheels appear to turn backwards.

The frame length of 3650 means 10 frames per day, i.e. 10 frames per rotation around the earth's axis. This should be enough to show how the rotation interacts with the revolution. The compromise, however, is that we can only show part of a year or we will have a very long movie (2 minutes!). After completing the animation, we can always scale the tracks in the Timeline to get the results we need for a particular use.

At frame 0, add a heading keyframe (r .H) for Year with a value of 0° using the Coord page of the Attribute Manager. At the last frame (3650) place another key and set the heading value to 360°. Drag the Timeslider to be sure the earth orbits once over the entire time of 3650 frames. Return to frame zero.

Daytime

Now let's rotate the earth.

In the Timeline, drag a copy of Year's rotation track down to Earth. For the key at frame 3650, change the heading to ?.

Think and predict before looking at the fine print below.

*The earth has to rotate 360° every day. Since it takes 365 days to go around the sun (revolution) it will have to rotate a total of 360 X 365 degrees. CINEMA 4D will do the math for you if you enter 360*365.*

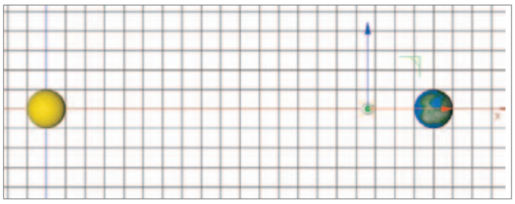
Move in on the earth and play a few frames to check out the motion.

The earth should both rotate (rapidly) and revolve (slowly) counter-clockwise, just as it does in our solar system when viewed from Polaris (the North star).

The moon

Make a moon (Moon) with a radius of 27.2 Create a moon material (moon) using the texture "Moon.tif" and apply in to Moon.

This radius is the correct proportion with regard to the earth. You may want to enlarge it a bit later. We also want to be able to revolve the moon separately from its rotation.



Create a new Null Object (Month) and make it a Child of Earth. Place it at the earth's origin as follows: While Month is selected, choose Functions / Transfer from the main menu. In the Object Manager, drag Earth into the Transfer to text box in the Attribute Manager. Make Moon a Child of Month and move it to an X position of –300.

Remember, this position is calculated from the Parent object, in this case, Month. The moon is actually a distance of 1700 units along the positive X-axis from the sun.

Using the time slider, scrub forward a few frames.

This is not correct. The moon is revolving far too fast. It is revolving as fast as the earth is rotating. Why? This is because the rotating earth is its parent and therefore it inherits its motion. However, it still needs to revolve around the sun with the earth. Since this should take a year, we should have made Month a Child of year.

Return to frame 0, make Month a Child of Year and move the Timeslider back and forth.

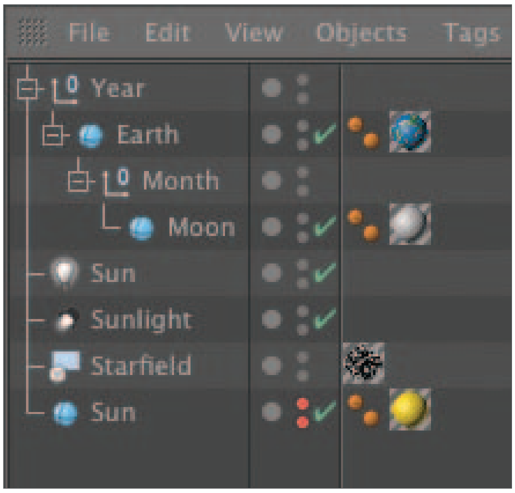
That's better, but now the moon is revolving only once per year (365 days). There are 28 days to the lunar month and 360 degrees of revolution per month. But how many months are in a year? Even though there are twelve months on our calendar, there would be approximately 365/28 = 13 lunar months in a year.

Copy the Rotation track from Year down to Month. Change the heading of the last frame accordingly . Play enough frames to check your work. Now move in close enough so see how the moon rotates as it revolves around the earth.

You should notice that the same side of the moon is always facing the earth, just as in reality. If this were not true, we would have to add a rotation track to the moon itself. Your completed hierarchy should look similar to that at the right.

Final steps

You have completed the technical part and here are a few suggestions for rendering.



• Create a material with only the luminance channel active and containing the *Starfield* shader. Apply it to a *Background* object.

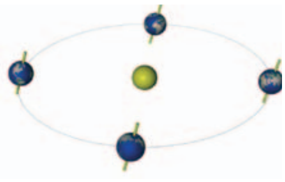
• Replace the Sun with a light for illumination. Keep its color close to white (the earth and moon textures have already taken the sun's yellow into account) and give it soft shadows with a density of 80. You will need to increase the *Map Size* if you want to see shadows from eclipses (and maybe reduce *Sample Radius*).

• Add a *Visible* light with *No Light Radiation* to simulate the sun itself. In the *Visibility* tab, adjust the *Inner* and *Outer Distance* to lend the sun size and also pump up the *Brightness* to add opacity (blocking out the stars).

• Copy the texture maps from the color channel into the luminance and bump channels of both the moon and earth materials (for earth's bump channel, use the "Earth.b&w" texture). In the luminance channels, minimize the *Brightness* and the *Mix Strength* to give a hint of starlight shining onto the dark sides of the earth and moon. Make the specular wide and low.

• Set up your camera and render no more than 400 frames at a resolution of 640 X 480.

A final word



You have been shown some of the facts about the relationship between the sun, earth and moon but we have overlooked a lot of details. The plane of the moon's orbit around the earth is tilted 5° to the plane of the earth's orbit around the sun. Otherwise, we would have eclipses twice a month. The axis of the earth's rotation is also tilted 15° to the plane of the earth's orbit around the sun.

This is what gives us the seasons. If you have time, here is a serious challenge. Add a Null Object that will tilt the axis of the earth 15° in such a way that it always pointing the same direction relative to the World system (figure at the right).

Creating an atmosphere

If time permits, you can experiment with materials to simulate an atmosphere. This is a basic technique and suitable for the level of detail of the earth's texture map.

Create a new primitive sphere (atmos), change its radius to 106 and make it a Child of earth. Create a new material (atm) and apply it to atmos. Activate the transparency channel and use the Fresnel shader with its default settings. Set the Mix value to 50%.

Fresnel (pronounced fren-**ell**) effects refract and reflect light differently from transparent surfaces, depending on the viewing angle. Without this option, your atmosphere's boundary will be too abrupt at the horizon. The Fresnel option will help make the atmosphere appear to thin out at its highest altitude. Another problem with our atmosphere is that it will tend to "wash out" the texture of the earth's surface below.

Also use the Fresnel shader in the diffusion and alpha channels. Make sure Soft is enabled (and Invert is unchecked) in the alpha panel. You can also use Fresnel in the color channel with Mix set to Multiply.

This will help "hold out" the atmosphere over the central region while allowing it to show at the edges.

Turn off the specular channel and pump up the Brightness in the color channel to 200. Move in close on the earth and render a region near the edge (make sure to use Best antialiasing from here on).

You could make the atmospheric effect more dramatic by increasing the brightness, but the earth's surface beneath would become more washed out. Again, without resorting to more elaborate methods, this should suffice for the present application. Now for some clouds.

Make another sphere (clouds) with a radius of 102, also a Child of earth (an earth-child so to speak). Create another material (cloud) also using the Fresnel shader in the transparency channel. This time, set the Mix type to Add and the Mix value to 90. Set the Brightness to 4.

The Fresnel will prevent the clouds from making the horizon of our atmosphere appear to have a hard edge. Using *Add* for the mix type and a *Brightness* of 4 will help give the clouds a bit of transparency in places other than just the edges.

Set the Brightness in the color channel to 100 (these are the clouds). Activate the alpha channel and apply the Cloud shader. Edit the Cloud shader: Color 1 = black, Color 2 = white, U Freq = 2, V Freq = 2, Clouds = 30. The options Soft and Image Alpha should be checked.

Here you are using the alpha channel to create the clouds (colored by the color channel) by eliminating the cloud-free areas.

In the specular channel, set the Width to 100 and the Height to 50. Apply the material to the clouds object and render.

One problem with this basic approach is that this cloud shader was meant for use in scenes on the earth's surface. It is too uniform for a global application. We cannot achieve the swirling patterns indicative of global weather without a substantial increase in effort. We can however break up the clouds so they do not uniformly cover the whole earth, with the help of one additional material. This will require the creative use of alpha.

Create a new material cloud2. In the alpha channel, use the Noise shader. Check Soft, Image Alpha and Invert. Edit the shader, setting the Global Scale to 400 and choose the Noise type Displaced Turbulence. Also in the shader's editor, set the Brightness to 4 and the Contrast to 80.

We are going to use this alpha to cause some of the regions of clouds in the other cloud texture to disappear.

Turn off all other channels. Activate the transparency channel and, yet again, select the *Fresnel* shader. Set the *Mix* type to *Add* and the *Mix* value to 100. Also, set the *Brightness* to 100.

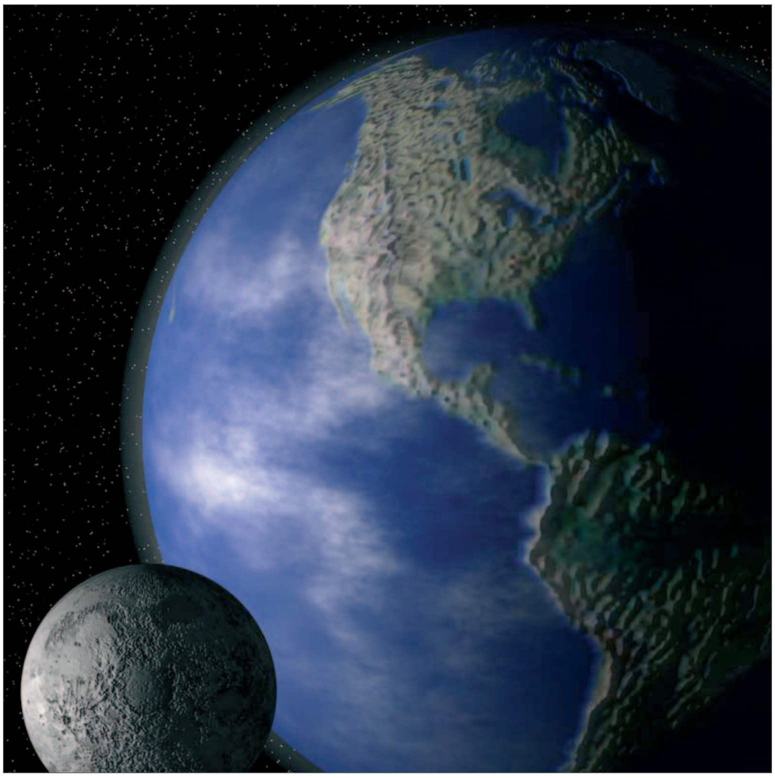
Like all alphas, the darker areas will be transparent, allowing the underlying textures to show through. We will apply this material in such a way that the clouds will be seen through these regions. The white areas of the alpha are the regions where the texture (cloud2) will show over the underlying texture (cloud). Since this overlying texture (cloud2) has only transparency (in addition to the alpha), it is this transparency that will replace any regions beneath the white areas of the alpha. Some of the clouds in the texture below will be replaced with transparency.

Apply the new material to the clouds object making sure it is to the right of the first material (cloud).

Notice that this process only works if the cloud object is separate from the earth object. Otherwise, the transparency would cause the earth texture to disappear in some areas allowing the stars to show through.

Move in close on the earth and render.

These textures can also be animated. You would probably want to do this mostly with cloud2 since it shows the larger, global patterns. Too much cloud activity will most likely make the motion jittery.



Introduction to the Motion System

In this tutorial we will use some of the tools provided by MAXON to create Non-Linear Animation (NLA) in CINEMA 4D. Non linear animation may already be familiar to some users from the field of video editing. NLA allows individual animation segments, so-called "Motion Clips", to be mixed in various ways. These Clips can be faded in or out, be looped any number of times, moved and scaled temporally and modified via curves.

First, create a new blank CINEMA 4D document and save it into your student directory as MotionSystem.c4d. In the project settings, ensure that the project has a duration of 200 frames.

We are going to create a bouncing ball that can be used within the NLA system. We could do this via the conventional keyfame animation system but using the Motion System method will give you much more flexibility in subsequently modifying the animation and reduce the work required to create a relatively complex animation.

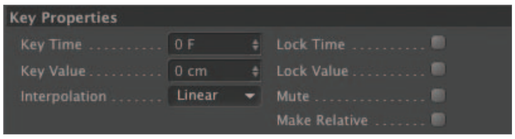
Create a Sphere primitive from the Objects menu at the top of the CINEMA 4D interface. In the Animation menu, enable keyframe recording for Position and Rotation parameters only and add a keyframe for the Sphere at Frame 0.

We are about to create a roll animation but we have included a Position keyframe to enable us to use another object later on, called a Pivot object. If we do not include a position keyframe the Pivot object will not behave as planned.

Move the Timeline to frame 20 and, in the Coordinates Manager, set the "P" Rotation of the sphere to 360º in the Coordinates Manager. Create a new keyframe for the sphere.

We have just created the roll animation. However, the keys created have a Spline interpolation causing the roll to speed up at the beginning and slow down at the end. This must be changed so that we can loop the roll later.

Select the first keyframe in the Timeline so that it turns orange. In the Attribute Manager, make sure Mode is set to Timeline. In the Key Properties, set Interpolation to Linear.



Key Properties

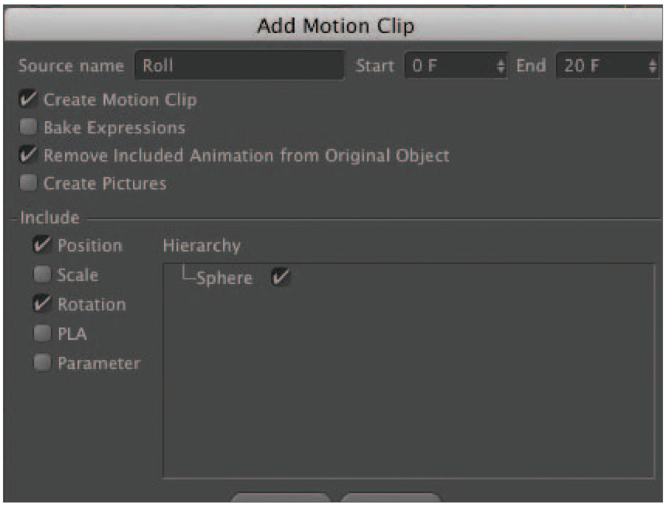
Key Time	0 F	Lock Time	□
Key Value	0 cm	Lock Value	□
Interpolation	Linear	Mute	□
		Make Relative	□

The motion of the Sphere is now constant. However, when we deselect the Sphere we can no longer tell that it is rotating. A texture needs to be applied to the sphere to help us see the rotation.

Double-click in the Material Manager to create a new material and name it "Ball". In the Color channel, in the menu next to the word Texture select Surfaces/Checkerboard. Drag the Ball material from the Material Manager and drop it on the word Sphere in the Object Manager.

Now we can see the Sphere rotate. Next we will create a Motion Clip of the rolling animation. We will create the Clip between frame 0 and frame 19, ensuring a smooth loop.

With the Sphere selected in the Object Manager, choose Add Motion Clip from the Animate Menu at the top of the CINEMA 4D interface. In the dialog box that appears, name the Clip "Roll". Set the duration to start at frame 0 and end at frame 19. Ensure that the Create Motion Clip and Remove Animation from Original Object options are checked. Also, make sure that in both Rotation and Position are selected the Include section. Press OK.



Add Motion Clip

Source name Roll Start 0 F End 20 F

☑ Create Motion Clip
□ Bake Expressions
☑ Remove Included Animation from Original Object
□ Create Pictures

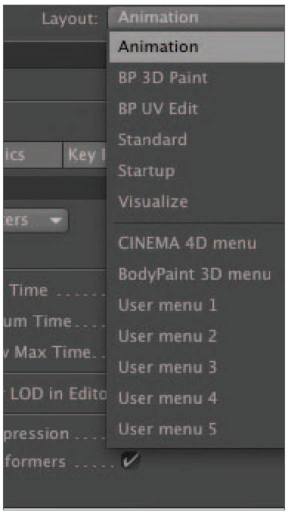
Include

☑ Position
□ Scale
☑ Rotation
□ PLA
□ Parameter

Hierarchy
└Sphere ☑

We have created a Motion Clip. The Sphere now has a new Tag attached to it in the Object Manager. This is a Motion System tag. We can see the Roll Clip if we change our layout to the Animation preset.

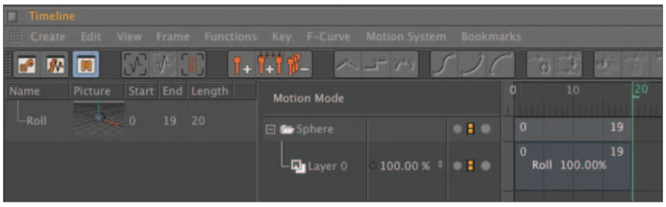
Switch to the Animation Layout. Select Motion Mode In the Timeline's View menu.



The original animation information has been removed from the sphere and is now contained in the Motion tag. Although it is now primarily used in the Motion System timeline, the source animation can be edited in both Key and FCurve modes if necessary.

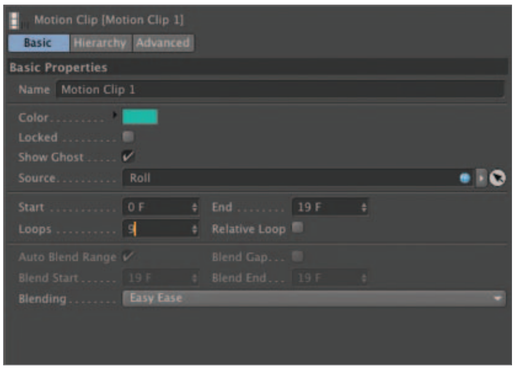
At the left of the Motion System timeline window is a preview image of our "Roll" Clip with the Clip's name next to with details of its length. On the right is the Clip placed between frame 0 and frame 19 in the Motion System timeline. You can see that it has been automatically applied to the Sphere in an Animation Layer. We will create another Animation Clip that will also be applied to the Sphere, but on a second layer. Between the words Layer 0 and the Timeline is percentage figure and an image of a Filmstrip. The percentage figure can be animated to vary the amount of influence a layer has on the overall animation. The Filmstrip is a toggle button allowing you to turn each layer on or off.

The Clip in the Motion Timeline is very flexible and can be modified in a number of ways, including scaling and looping.


Timeline
Create Edit View Frame Functions Key F-Curve Motion System Bookmarks
Motion Mode
Sphere
Layer 0
100.00 %
Roll 100.00%

Name	Picture	Start	End	Length
Roll		0	19	20

Click on the Clip in layer 0 to select it. In the Attribute Manager, select the Basic tab and set Loops to 9.

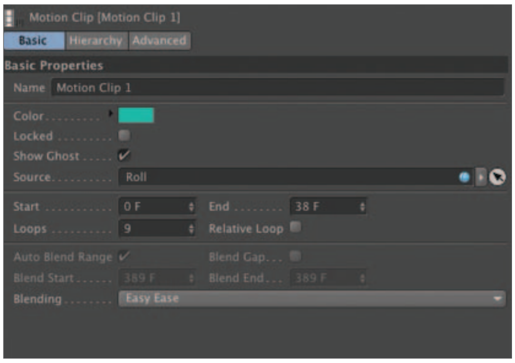


Name Motion Clip 1
Color
Locked □
Show Ghost ✓
Source Roll
Start 0 F End 19 F
Loops 9 Relative Loop □
Auto Blend Range ✓ Blend Gap □
Blend Start 19 F Blend End 19 F
Blending Easy Ease

The animation is now set to repeat 9 times after the original Clip. If you press the Play button now you will see the Sphere roll for the entire duration of the project. You can also modify the speed of the rotation by scaling the Clip's duration.

In the Attribute Manager, set End to 38.

The rotation will now be interpolated over 38 frames and will take twice as long to complete. You will also notice that the "Roll" Clip in the Timeline shows it has been scaled by 200%. Try some different numbers to see the results.



Name Motion Clip 1
Color
Locked □
Show Ghost ✓
Source Roll
Start 0 F End 38 F
Loops 9 Relative Loop □
Auto Blend Range ✓ Blend Gap □
Blend Start 389 F Blend End 389 F
Blending Easy Ease

When you have finished experimenting, reset End to 19 (make sure that the current frame marker is at frame 0). Mute Layer 0 by clicking on the filmstrip icon.

Now we will create a bounce animation to add to the Clip library.

In the Timeline toolbar, make sure that only Position keyframes are active. Select the Sphere and record a keyframe for the Sphere at frame 0. Move the timeline to frame 10 and, in the Attribute Manager, switch to the Coord tab. Set the Y Position to 500. Record a keyframe at frame 10. Move to frame 20, set the Sphere's Y position to 0 and record another keyframe.

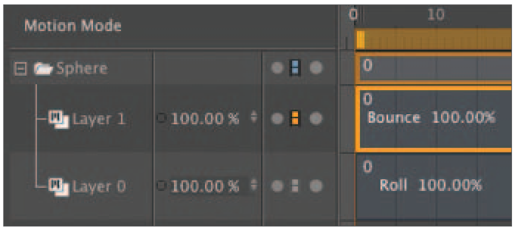
Now we have a single bounce, which we can use to create another Motion Clip.

With the Sphere selected in the Object Manager, choose Add Motion Clip from the Animate Menu. In the Add Motion Clip dialogue box, name the Clip "Bounce". Set Start to 0 and End to 19. Disable the Create Motion Clip and Remove Animation from Original Object options. Also make sure that Position is selected in the Include section. Press OK. Right-click on "Sphere" in the Timeline's "Motion Layer" window and add a new layer (Add Motion Layer). Drag the newly created "Bounce" clip onto the Keyframe window next to Layer 1.

As you can see, another layer has been added to the Motion System timeline. This can be manipulated exactly the same as the previous layer.

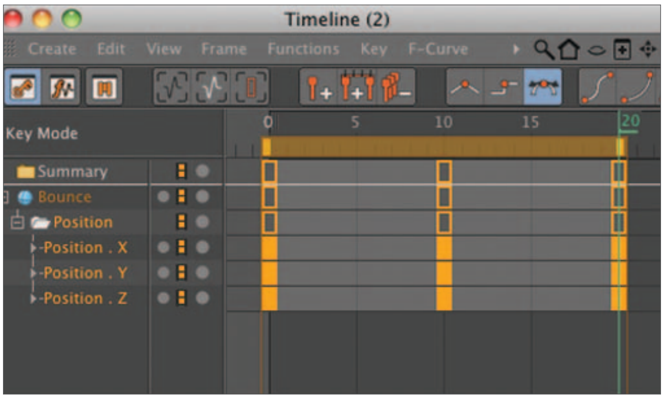
Click on the Clip in layer 1 to select it. In the Attribute Manager, select the Basic tab and set Loops to 9.

The animation is now set to repeat 9 times after the original Clip. If you press the Play button now you will see the Sphere bounce for the entire duration of the project. However, the Sphere doesn't appear to bounce convincingly because it decelerates as it approaches the bottom of the bounce. We must therefore edit the source animation data.



Right-click on the Clip in Layer 1 and select Show in Key mode from the menu.

This will open a new Timeline view in a separate floating window. This is where we can access the animation data defining the "bounce" Clip.



Select the Y position keyframe at frame 0. In the Attribute Manager for the selected key, disable Auto Tangents and enable Zero Length. Do the same for the keyframe at frame 20. Close the dialog window.

We now have a pretty convincing bouncing ball, but it doesn't do much more than bounce on one spot. To move it around we can use the Pivot object from the NLA toolset. The Pivot object was especially designed for use with the Motion System to interactively move, scale and rotate an animation within a Motion Clip.

Go to frame 0. unmute Layer 0 by clicking on the grayed-out filmstrip and click on the Clip on Layer 0 to select it. In the Attribute Manager, switch to the Advanced tab. Click on the Create Pivot button.

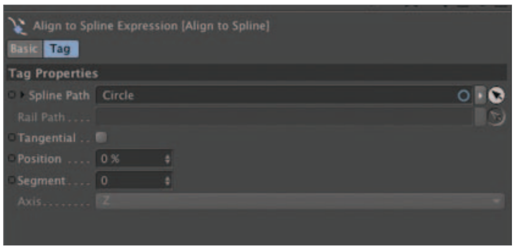
We now have control over the location of the bouncing ball via the Sphere Pivot object. Play the animation and move the object around the viewport to see the results.

Create a Circle Spline from the Spline Primitives menu. In the Attribute Manager's Object tab, set Radius to 800m and Plane to XZ. Select the Sphere Pivot object in the Object Manager and, in the Tags menu, select CINEMA 4D/Align to Spline.

We will now keyframe the Sphere Pivot object to travel along the Circle Spline.

In the Object Manager, make sure that the Align to Spline tag is selected and that the tag properties are visible in the Attribute Manager. Click and drag the Circle Spline into the Attribute Manager and drop it onto the Spline Path field. Make sure that you are at frame 0 in the Timeline and Ctrl+click on the hollow circle next to the word "Position" to record a keyframe for this parameter.

We have just created a keyframe for the position parameter of the Align to Spline Tag. This is indicated by the solid red dot next to the Position Parameter.


Align to Spline Expression [Align to Spline]

Basic Tag

Tag Properties

Spline Path Circle

Rail Path . . . .

Tangential . . □

Position . . . . 0 %

Segment 0

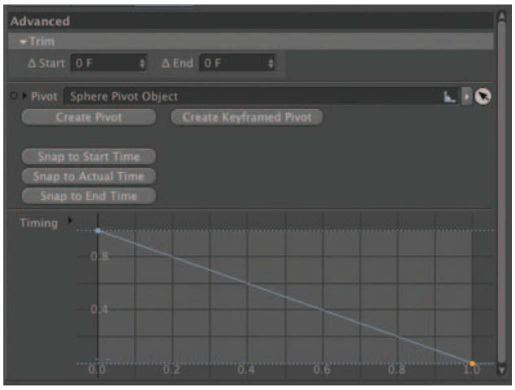
Axis . . . . Z

Go to the last frame of the project. Set the Position parameter to 100%. Ctrl+click to set a keyframe. Switch the Timeline to Keyframe mode and select the keyframe for the Align to Spline tag at frame 0. In the Keyframe Attributes, set Interpolation to Linear.

When you press the play button now you will see the sphere bounce around the circle. However, the rotation of the sphere does not follow the circle's direction.

Make sure the Align to Spline Tag is selected and in the Attribute Manager enable the Tangential option.

Now the Sphere Roll rotation follows the Circle's path but the Sphere appears to be Rolling in the opposite direction to the movement around the Circle.



Advanced

Trim

Δ Start 0 F Δ End 0 F

Pivot Sphere Pivot Object

Click on the "Roll" Clip on Layer 0 to select it. In the Attribute Manager, switch to the Advanced tab and make the Timing graph is visible. Select the point to the left of the line and position it at the top left corner of the graph. Select the point at the right end of the line and position it at the bottom right corner of the graph. The line should now slope down to the right.

We have just reversed the timing of the Roll sequence and the Sphere will now rotate in the correct direction while bouncing around the Circle spline.

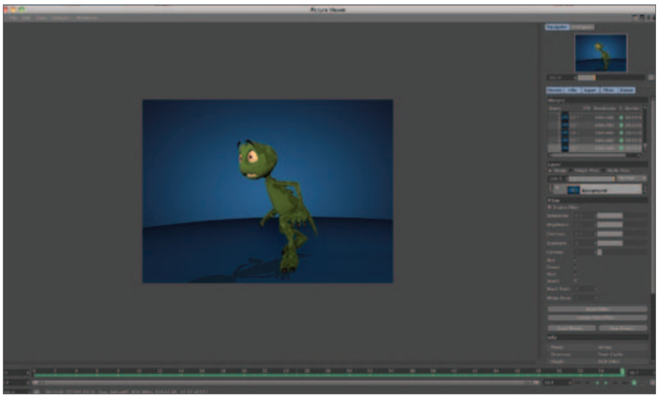
Introduction to Rendering

Overview

Rendering is the final act in putting all the pieces together. At its core, computer animation is all about numbers representing the positions of points that define the surfaces of objects. More numbers are involved in specifying what a surface should look like at any given point through texturing and lighting. Even more numbers are involved in calculating how these modeled, textured and illuminated surfaces should move or be animated. The final act of crunching all these numbers to produce the effect of an image on the screen is the act of rendering. Rendering can produce a range of results from low quality, quickly rendered images to smooth, finely detailed images of a high resolution still or animated sequence. The output of a rendering can be put out in a variety of formats and sizes depending on the final use. In addition to the final realization of all the modeling, texturing, lighting and animation decisions you have made throughout the process, the rendering stage is where you will finally see the results of effects such as motion blur, depth of field and special lighting techniques.

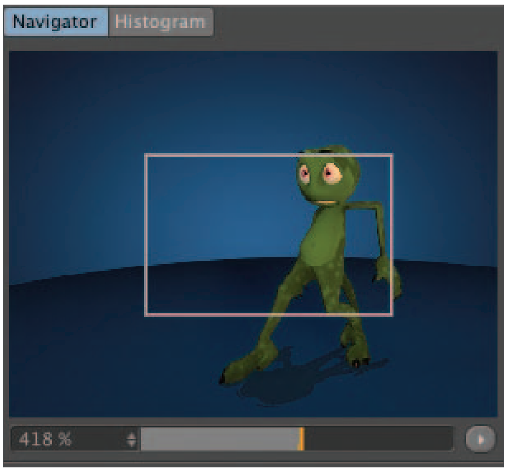
The continuing development and optimisation of the render engine in CINEMA 4D enables us to render faster using less memory. For example, the CINEMA 4D render engine uses a technique called Bucket Rendering. This provides a tremendous improvement in memory efficiency. Other improvements have resulted in many render-intensive tasks becoming much faster. As a result we can now create scenes of great complexity rendering billions of polygons on relatively modest hardware.

The Picture Viewer



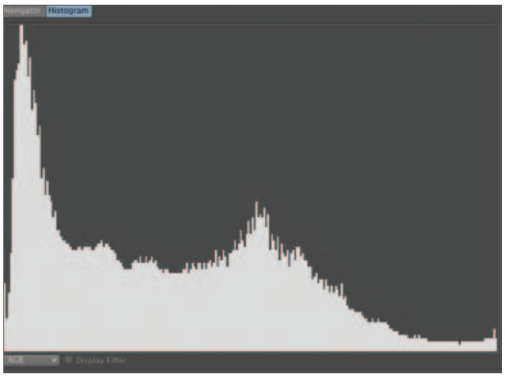
So far, all of your rendering has been done in the editor. This lets you see the results of your progress, especially with regard to lighting and texturing. Rendering a region can also serve as a quick check of different material or illumination settings in an important area. However, if you want to save a rendered image or be able to see certain effects you must render to the Picture Viewer.

The Picture Viewer is a window that shows the rendering as it progresses and was given additional functionality in CINEMA 4D R12. It can be closed to allow work to continue in the editor while rendering continues in the background. You can even leave CINEMA 4D to work in another application while the Picture Viewer is rendering.



The majority of the Picture Viewer's interface is taken up by the current render, but the Picture Viewer also displays other useful information. At the bottom of the window are the pixel dimensions of the currently displayed image and an indication of how long it took to render. To the right of the window is a panel showing more information, including the name of the file from which the rendered image was created and the amount of memory required to render it. At the top of the panel is the Navigator. This displays the current rendered image in preview. Depending upon the zoom factor of the main window a moveable square will indicate the viewable area within the main render view. Double click in the Navigator to set the zoom to 100% in the main render window or select a region to view by Control clicking.

The Picture Viewer also allows you to perform useful functions such as comparing previously rendered images to the current render using the *History* tab. You can also specify two previously displayed images to be viewed on screen simultaneously with a movable boundary line allowing you to 'AB Compare' the two. These functions are especially useful when making subtle alterations to an image. The AB Compare function can help you judge whether your latest adjustments to your rendered scene have actually resulted in a better image.

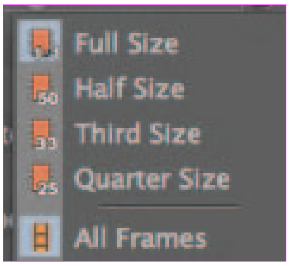


The Histogram displays a static representation of the distribution of brightness, color and luminance values within the currently displayed image. The horizontal axis represents the brightness values from darkest on the left to brightest on the right. The vertical axis shows how much of the image is found at any particular brightness. The Histogram can help you judge at a glance whether an image is correctly exposed or if its contrast is correct among many other things.

Using the Mode selection menu you can chose which value or color channel to display in the Histogram.

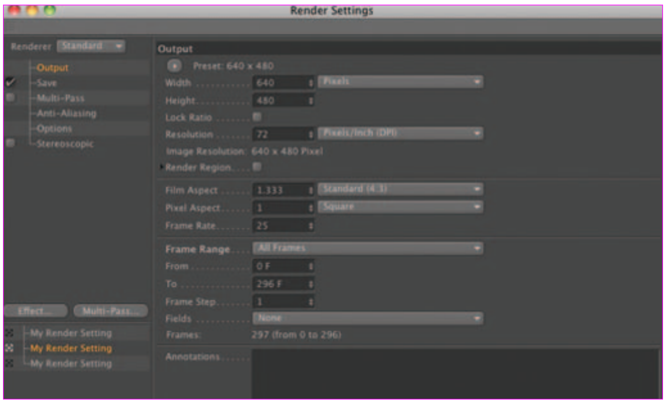
With the exception of a histogram that is bunched up to the right hand side (over exposed) there really is no such thing as a 'bad' histogram or even a 'good' one. The histogram just shows you how things are, it's up to you whether you need to make changes.

The Picture Viewer is also able to display animated content as well as still images. The integrated RAM player lets an animation be played directly in the Picture Viewer without first having to save it as a video. This is done by loading an animation's rendered images into virtual memory so they can be played back in real-time.



If an animation or video is loaded into the Picture Viewer a control panel similar to the compact Animation Palette will be displayed at the bottom of the Picture Viewer's dialog window. Successfully cached images are displayed in green in the Timeline and if the sequence of images is played and the Timeslider (the green rectangular marker) moves beyond this green range new images (from the hard drive) will be loaded and the oldest images will be removed - which rarely works without slowing the frame rate of the animated material.

As the number of frames that can be cached is dependent upon the amount of physical RAM available, there are also a number of options available to avoid the Timeslider attempting to display un-cached images. These include the option to scale down high resolution frames and the ability to select a limited range of frames to preview.



Render Settings

Renderer Standard

- Output
- ✓ Save
- Multi-Pass
- Anti-Aliasing
- Options
- Stereoscopic

Effect... Multi-Pass...

- My Render Setting
- My Render Setting
- My Render Setting

Output

Preset: 640 x 480

Width 640 Pixels

Height 480

Lock Ratio □

Resolution 72 Pixels/Inch (DPI)

Image Resolution: 640 x 480 Pixel

Render Region □

Film Aspect 1.333 Standard (4:3)

Pixel Aspect 1 Square

Frame Rate 25

Frame Range All Frames

From 0 F

To 296 F

Frame Step 1

Fields None

Frames: 297 (from 0 to 296)

Annotations

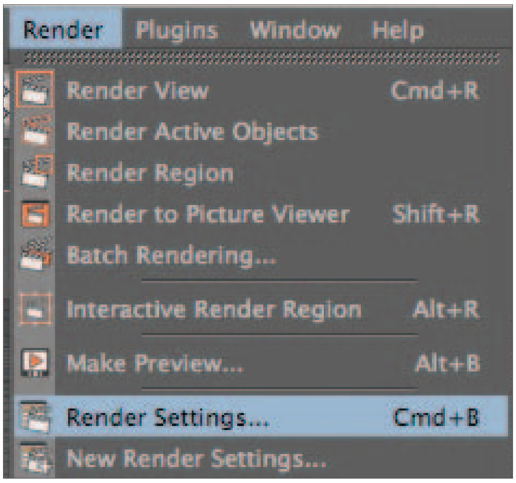
It should be noted that not only are you able to view images created in CINEMA 4D but the Picture Viewer is also able to open digital images and animations saved in most common formats.

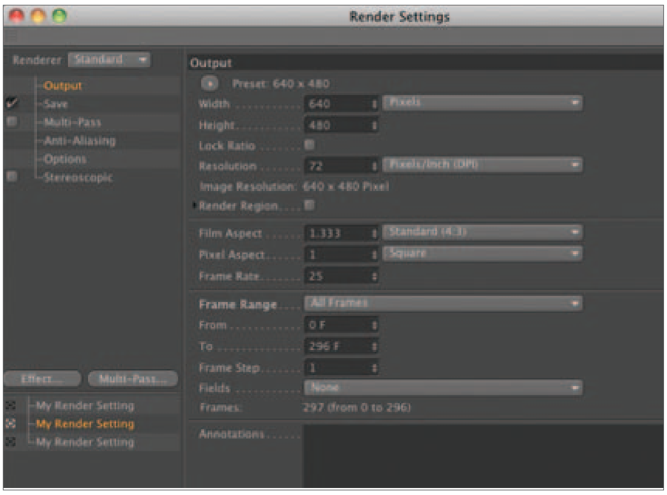
Although most parameters specifically affecting the rendering are set prior to beginning a render in the *Render Settings*, there is also a *Filter* tab in the Picture Viewer that allows you to make adjustments to parameters, including Brightness, Contrast and Saturation, after the image has been created. Once you are satisfied with any adjustments you have made using these adjustment controls the image can be saved in its altered form and you can also save the settings as a preset Post Effect for use on subsequent renders, if needed.

The remainder of this introduction refers to parameters and features that are set and adjusted in the *Render Settings*.

Antialiasing

Without **antialiasing**, object edges have a jagged, stair-step appearance. Images are displayed on televisions and computer monitors using tiny square points of color (pixels). You can see the effects of this "pixilation" if you look closely at curved letters on your monitor (depending on your operating system). Antialiasing is the process of smoothing these lines and curves. In antialiasing, an area around each pixel is evaluated to see how the color changes in that area. A new color is then calculated for each pixel being evaluated and blends it smoothly with the rest. This means that the stair-steps can be filled in with colors between the object's edge and the background, thereby softening the appearance on the screen. The size of the area evaluated can be user-defined. A 1 X 1 area means only the one pixel is "evaluated". In other words, no antialiasing will take place. Higher values, up to 16 X 16, give progressively smoother results at the cost of greater render times.




Render Settings

Renderer Standard

- Output
- ☑ Save
- □ Multi-Pass
- Anti-Aliasing
- Options
- □ Stereoscopic

Effect... Multi-Pass...

- My Render Setting
- My Render Setting
- My Render Setting

Output

Preset: 640 x 480
Width 640 Pixels
Height 480
Lock Ratio □
Resolution 72 Pixels/Inch (DPI)
Image Resolution: 640 x 480 Pixel
Render Region □
Film Aspect 1.333 Standard (4:3)
Pixel Aspect 1 Square
Frame Rate 25
Frame Range All Frames
From 0 F
To 296 F
Frame Step 1
Fields None
Frames: 297 (from 0 to 296)
Annotations

There are two principal types of antialiasing. The antialiasing of object edges, where they are silhouetted against the background, is called *Geometry* in CINEMA 4D. This is the default mode and works well for many stills and most animation where movement is rapid or the format is small. The other mode is called *Best* and antialiases color transitions in textures, shadow edges and the area behind transparent objects. *Best* antialiasing includes object edges (*geometry*). This is the mode to use for high-quality work - especially stills and objects with detailed surface characteristics.

You also have a choice of filters to use with antialiasing. *Still Image* gives sharp edges for high-quality still images. The *Animation* filter works best with animation: By softening the edges somewhat it reduces the jittery motion sometimes produced when animating objects.

Although antialiasing should be used on almost all final output, you may want to disable it when testing motion, scene composition or other test scenes in order to save render time. The antialias setting also affects rendering in the editor. You can disable it temporarily or, better yet, create a separate set of render settings for use while working on your scene in the editor.

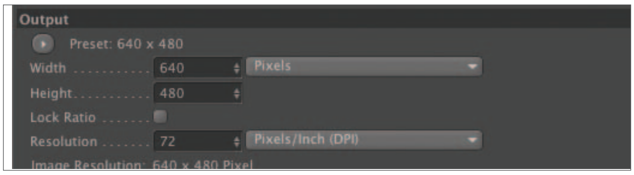
Resolution

Before rendering, you must decide the size of your output. This is always measured in number of pixels wide by number of pixels high. Higher resolutions can offer more detail and clarity but consume more memory and take longer to render. On your screen, a higher resolution rendering merely results in a larger image. However, when printed or output to film, a larger image will have finer detail when shrunk down for final use. If you are looking at the two renderings to the right ("Round") on screen, the bottom one may appear a bit sharper in detail but in print the difference is very notable. The bottom one was rendered at four times the size of the upper one and then saved at 300 dots/inch as compared to 72 dots/inch for the upper rendering.

Round
Round

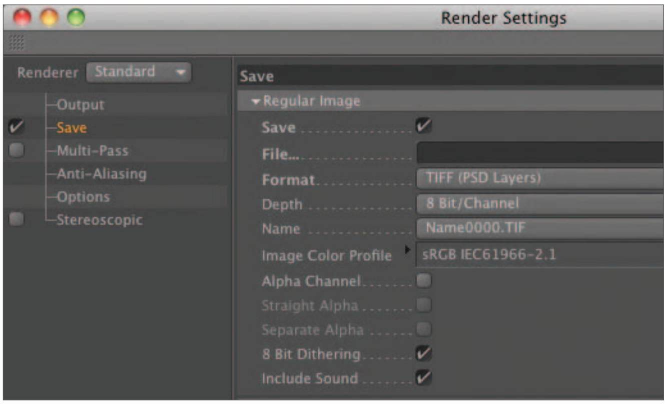
There are several standard sized formats depending on the destination of the output. For animations these range from 320 X 240 (MPEG web animations), 640 X 480 for video in North America to 2048 X

1536 for commercial film (there are other formats associated with these uses as well and a table listing these can be found in the Help files). For still work the resolution can vary from very small (web graphics) to 1024 pixels (for a computer monitor background) or even 4096 pixels or larger for printed output. Frame rate is also defined in the render settings, independently of that which is defined in the project settings. The default value of 30 frames per second (actually 29.97) corresponds to standard video output (in North America).



Output
Preset: 640 x 480
Width 640 Pixels
Height 480
Lock Ratio □
Resolution 72 Pixels/Inch (DPI)
Image Resolution: 640 x 480 Pixel

Output formats



Render Settings
Renderer Standard
- Output
- ✓ Save
- □ Multi-Pass
- Anti-Aliasing
- Options
- □ Stereoscopic

Save
Regular Image
Save ✓
File...
Format TIFF (PSD Layers)
Depth 8 Bit/Channel
Name Name0000.TIF
Image Color Profile sRGB IEC61966-2.1
Alpha Channel □
Straight Alpha □
Separate Alpha □
8 Bit Dithering ✓
Include Sound ✓

There are many different systems and methods for compressing and storing graphic data. Only the three most important for our purposes are listed here. Some compressors save all information and quality put out by CINEMA 4D. These are referred to as "non-lossy". Others will lose some fidelity ("lossy") when smaller formats are used.

JPEG is a standard image format for stills on the web. It uses a "lossy" compression which discards some information but can produce small file sizes that load quickly on the web. When using the QuickTime JPEG, the amount of compression can be adjusted for a balance between quality and file size.

TIFF, or *tagged image file format*, is a standard output for high-quality stills. It is also standard for the highest quality of animated output. However, when used for animation, a series of individual TIFF images will be rendered that must subsequently be linked in post-production using another program.

QuickTime Movie is the standard we will use for animation. It offers many compression options with varying degrees of quality. The best results are achieved using the Animation CODEC (Compressor, Decompressor) at its "best" setting. However, this can take up large amounts of memory so it must be used with care.

Once an animation has been rendered it can be opened in QuickTime and exported to other formats at the same quality or less. Since the render time is the same regardless of the CODEC used, the strategy you should use is to render at the highest quality, then export at lesser quality. Once you are sure the compressed file is satisfactory you can delete the larger, high-quality file or burn it to disc for storage. Another option associated with output is the dots per inch, or DPI. This is only meaningful if you are going to output the image for printing.

Other settings

Several additional rendering parameters can be set in the render settings. You can turn off transparency, reflection and shadows to speed up rendering in motion tests. You can adjust the range of frames to render, also useful in motion tests. You can also enter a name and location to which to save the rendered file.

Global Illumination, caustics, motion blur, depth of field and other more advanced effects also have parameters that are defined in the render settings. Certain aspects of the raytracing process can also be controlled here. Although this is generally not necessary, certain situations, such as reflections from many surfaces, may require adjusting these settings. You can read much more about rendering in the help files.

Rendering Intro Tutorial

This tutorial will serve as a brief introduction to some of the procedures and parameters of rendering. We will practice these using the indoor scene featuring the table we assembled during the lighting tutorial.

1. Render settings and saving files

In this section you will learn how to create new render settings and render to specific files. You will also render with different antialiasing settings for comparison.

Open the file you created for the indoor scene featuring the table, "Table Scene 1". Save the scene as "Table Scene 2".

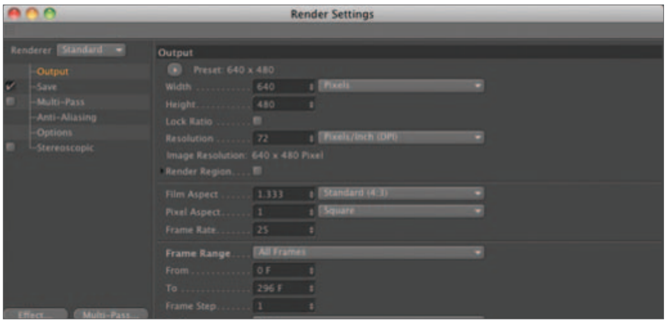
Any changes you now make can be saved to the new file, keeping the previous version intact.

Back in the "Table Scene" folder, create a sub-folder named "renders".

This folder is for organizational purposes. You are about to create several renderings for comparison. Storing them in a separate folder will keep the clutter in the parent folder to a minimum.

Back in CINEMA 4D, choose Render / Render Settings... from the main menu.

This will open the Render Settings menu and allow you to create a unique set of rendering parameters.

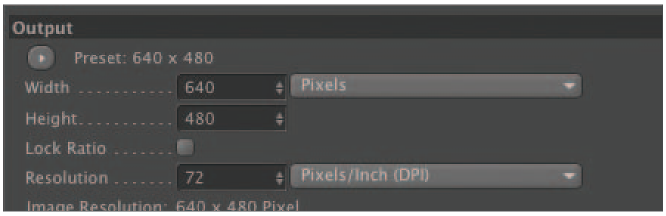


Render Settings

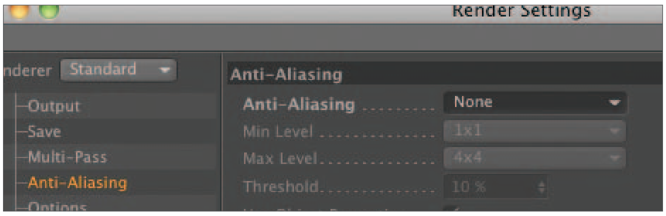
Renderer: Standard

Output
Preset: 640 x 480
Width: 640 Pixels
Height: 480
Lock Ratio: □
Resolution: 72 Pixels/Inch (DPI)
Image Resolution: 640 x 480 Pixel
Render Region: □
Film Aspect: 1.333 Standard (4:3)
Pixel Aspect: 1 Square
Frame Rate: 25
Frame Range: All Frames
From: 0 F
To: 296 F
Frame Step: 1

Double-click on 'Render Settings' at the bottom left of the dialogue and name the new render settings "No AA". In the output panel, set the Width to 640 and Height to 480; in the Antialiasing panel, set Antialiasing to None.



Output
Preset: 640 x 480
Width: 640 Pixels
Height: 480
Lock Ratio: □
Resolution: 72 Pixels/Inch (DPI)
Image Resolution: 640 x 480 Pixel



Render Settings

Renderer: Standard

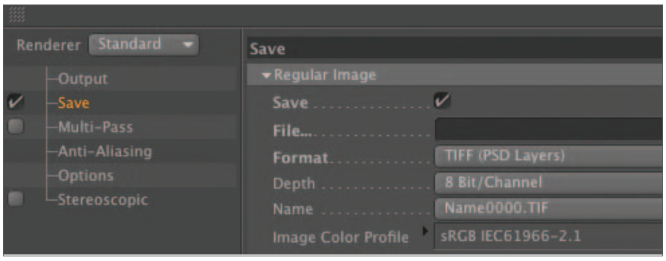
Anti-Aliasing
Anti-Aliasing: None
Min Level: 1x1
Max Level: 4x4
Threshold: 10 %

Now we need to declare a name for the rendered file and specify a place to save it.

In the Save panel, click on the Path. Navigate to the "renders" folder (in the "Table Scene" directory). Name the new render file "tableScene aa=None".

This is an uncommon type of naming scheme and is useful for testing. Alternatively we could have used "render tst-None", for example. Note that in both cases "None" is capitalized to indicate that it is an important characteristic of the rendering. This is an example of the importance of using some type of organizational system to track your experimentation.

While still on the Save page, make sure the format is set to TIFF (PSD Layers).



Renderer: Standard

Save
Regular Image
Save: ☑
File...
Format: TIFF (PSD Layers)
Depth: 8 Bit/Channel
Name: Name0000.TIF
Image Color Profile: sRGB IEC61966-2.1

All other settings can be left in their respective default mode. Before proceeding, draw a table on a fresh sheet of paper. The table should have four columns, the last taking up about 40% of the width of your paper. Name the first column "File name", the second "Render time", the third "File size" and the fourth "Comments". For your first entry, record the name "tableScene aa=None" in the first column.

Close the Render Settings dialog and select Render to Picture Viewer from main menu or by clicking on the corresponding icon in the top Icon Palette. Record the render time in your chart.

The render time is displayed in the bottom left corner of the Picture Viewer and in the bottom left corner of the C4D interface (hours:minutes:seconds).

Create a new set of render settings named "Geo". Set Antialiasing to Geometry and change the file name under which the image will be saved to "tableScene aa=Geo". Set the resolution to 640 X 480 and make sure the output format is TIFF. This should be the same as the "No AA" settings, except for the level of antialiasing and the name of the output file.

Render and record the render time in your table.

Now you have two rendered files that differ only in the amount of antialiasing each has. Let's add one last file for comparison. Create another new render setting using Best antialiasing. Also, set the resolution to 640 X 480 (maintain the TIFF format). Render a new file (name it appropriately) to the same folder as before and note the render time.

2. Antialiasing comparison

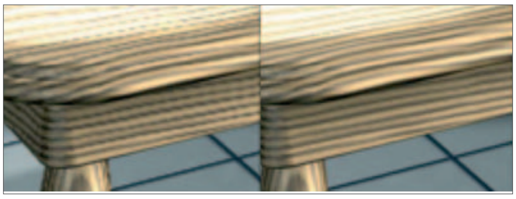
You now have three renderings using three levels of antialiasing. We will now compare these to see the effects of the antialiasing. CINEMA 4D is not useful for this because it only displays one rendered output at a time. This is not a problem. This is the job of such standard graphic utilities as Preview in OS X or the QuickTime Player in the case of movies (or stills).

Exit (but do not quit) CINEMA 4D by clicking on the narrow margin you have been leaving at the right of the screen. Navigate to the "renders" folder where you have stored the three rendered files. Open all three files with Preview or another image viewing utility on your computer.

If you are using Preview, drag the images onto the application one at a time. Otherwise they will be opened as a group and side-by-side comparisons will not be possible.

Move the opened images around the screen so that you can make a comparison.

Study the differences between the renderings. Think of them in terms of "none", "edge" and "all". The *Geometry* setting antialiases only the edges (silhouette) of objects whereas *Best* also antialiases the textures. Enlarge the images and compare sections of the wood grain (in the image to the right, *Geometry* is on the left, *Best* is at the right).



Frequently, especially during testing, you will want to overwrite the same file after making changes in response to viewing a render. When you attempt this, CINEMA 4D will ask if you want to overwrite the existing file. If you answer yes but CINEMA 4D responds that it cannot write to the file, it is because that file is open and in use by another program, probably the Picture Viewer or QT Player. Exit CINEMA 4D, close the rendered file and try again.

3. TIFF vs. JPEG

TIFF images offer high quality but with correspondingly large file sizes. Let's render a JPEG image for comparison.

Create a new render setting named Best.jpg. Adjust all settings to the same used for Best in the TIFF renders above except for changing the file format to JPEG instead of TIFF. Render as "table scene Best.jpg". Open this file and the TIFF version of Best in the Preview application.

At first glance these two images may appear to look the same. Let's take a look at two major differences.

While in the Preview application, select each of the two images and make each double size. Rearrange them on the screen so you can make a comparison, especially of the vase.

If you look closer, you will probably notice the higher quality of the TIFF image. The difference would be more noticeable in a printed image. Generally speaking, use TIFF if for rendering high-quality images for print. So why not use TIFF all the time?

Navigate to, and open the "renders" folder. With the folder selected, make sure that View / As List is chosen from the main menu (on a Mac). Widen the folder to the right if necessary to view the file sizes.

The TIFF images should all be substantially larger than the JPEG image. Make a note of the sizes, names and render times of your table.

4. QuickTime movies

The last part of our introduction will focus on the creation of an animation in QuickTime format. QuickTime is a good all-around choice for animation output. QuickTime cross platform, widely used and offers many different choices of format and CODECs for compression. Let's animate the camera to move in for a close-up on the vase.

Rename the Object Camera Cam: still. Make a copy of it, name the copy Cam: motion and delete its Protection Tag. Select Cam: motion to view the scene through.

Our original camera was left untouched and is set up for still shots. The Protection Tag for the new camera was deleted so you can animate the camera.

Choose Cam: motion as the scene camera through which to view the scene in the perspective viewport. Also be sure Cam: motion is the selected object (with its name orange) in the Object Manager. At frame zero, with only the position and rotation tracks active in the Animation Toolbar, record keyframes. Move the Timeslider to frame 60.

Animating good camera moves can be a very tedious process. There are many tools and techniques you can use to get good results. Since the main concept here is rendering, we will not spend time perfecting the move, but let's at least visit one important consideration.

If you only use the Position Track to move the camera, it will have to move to the side as well as forward in order to bring the vase into the view's center. This will result in a paning type of movement. While there are uses for this type of movement, you will generally want a smoother, more flowing motion.

Before moving the camera's position toward the vase, rotate the vase into the center of view as closely as possible. Now move the camera in so the vase takes up most of the view height and record the keyframes. Run the Timeslider back and forth to make sure you have captured the animation and the look is satisfactory.

If the action is too slow when the slider is moved, change the *Level of Detail* to low in the *Display* item of the view menu. If you are ahead in the tutorial, set up another camera and record the same motion, this time without using a rotation track. Compare the look of the two motions in the editor. If your vase was not much off-center to begin with, the difference will be very subtle. However, this is a good habit to get into and will make a noticeable difference when moving in on objects at the edge of the field of view or completely off-camera. Now we will set up the render.

Create a new set of render settings named "motion tst".

There are many other names you could have used. This one tells us the render will be a movie instead of a still and, being a test, will probably not be antialiased for faster rendering.

Set Antialiasing to None and also set Shadow to None.

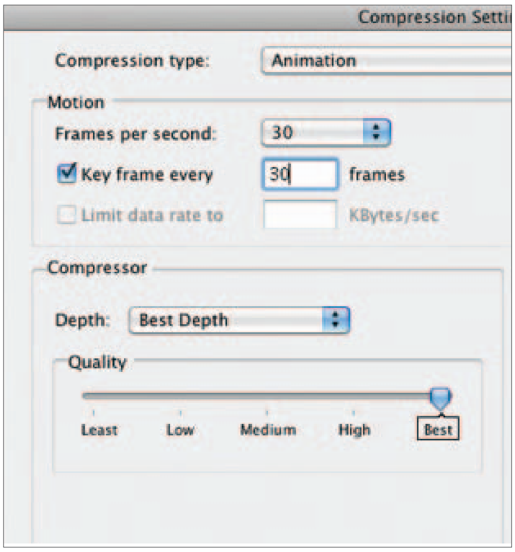
Since this is a test to see how the motion looks, we will speed it up as much as possible. For checking motion in larger, more complex scenes you may want to enable one of the *Render As Editor* options (general page). For additional speed, set the editor display to Wireframe. This is similar to playing the motion in the editor. However, since it is all calculated and recorded while rendering, it will play back more rapidly as a movie.

Set the output Resolution to 320 X 240.

This smaller format will also speed up the render.

Also in the Output panel, set the range of frames to be recorded by entering 0 and 60 in the appropriate input boxes after Frame. Name the output file "tableScene motion tst 1".

You could give it a name like "table test 1" to keep the name a reasonable length. Since the file will be appended with ".mov" once we select QuickTime Movie as the output, we will still recognize it as a motion render without the term "motion". The main goal of naming is to have a good idea what each file is.



Compression Setti

Compression type: Animation

Motion

Frames per second: 30

☑ Key frame every 30 frames

□ Limit data rate to KBytes/sec

Compressor

Depth: Best Depth

Quality

For the Format, select QuickTime Movie. Open up the Options dialog. The Compressor should be Animation and the Quality should be Best by default (also make sure that the lower of the two text boxes under Compressor reads Millions of Colors or Best Depth).

These high-quality settings are not necessary for a motion test. These values only affect the saved file, the render time is the same for any choice of compression. A higher quality compression choice will result in a larger file size but you will continue to write over the file until the motion looks correct. Also, being a test, the file will eventually be deleted. As already stated, it is generally best to render at the highest quality and then export to the final desired output quality using a post production application such as QuickTime.

Render your QuickTime movie. Note and record the time it takes to render the first frame.

While the rendering continues, calculate the approximate total rendering time based on the time to render the first frame. This will not always give precise results, especially if new objects enter the scene or special lighting effects activate.

After the render is complete, return to the desktop, and open the new movie file "table scene tst 1".

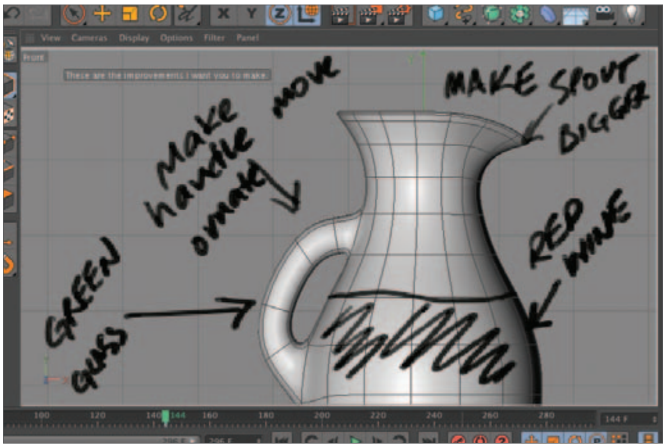
Since the file is a ".mov" file, it will automatically be opened by QuickTime if you double-click on it.

Play the movie.

Larger, longer or more complex movies may show jerkiness on playback due to frames being skipped in attempting to achieve the playback rate you set for the output. You can prevent this by selecting *Play All Frames* from the *Movie* menu item in QuickTime Pro. Of course this will slow the movie down to less than the desired speed but will give you a better idea of how the motion looks.

That is your first rendered animation! Make any changes to the camera move if you want. If you are happy with the motion and want to render at a higher resolution with shadows and antialiasing, create a new set of render settings. Serious improvements to the render settings will cause the rendering to run after class and even on into the night.

Working with the Doodle tool



This tutorial will show how the Doodle tool allows you to draw directly in the editor view making it easy to comment on work in progress, make notes, create diagrams and much more. These doodles can then be used by others for reference while working on the scene.

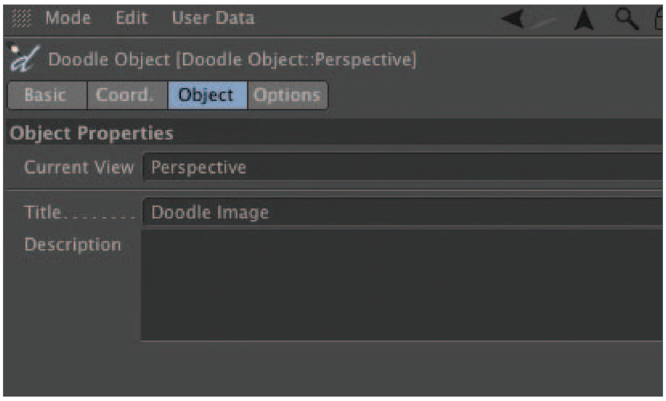
The best way to doodle is with a graphics tablet but a mouse can be used as well. Doodles are saved as a Doodle object (a Doodle object can only save the Doodles of a single Viewport). It is, however, possible to add a Doodle at different frames of an animation.

The Doodle functions in CINEMA 4D are, for example, very useful as a teaching aid. It can be quite difficult to understand and remember verbal instructions about how to improve a model or image. However, the Doodle tool lets both teachers and students communicate directly within the CINEMA 4D interface.

Create a new CINEMA 4D document. Activate the Doodle Paint tool by selecting it from the CINEMA 4D Tool menu. Click and drag with the mouse to paint in the Viewport

You should now see a white line where you just painted on the Viewport. You will also see a new Doodle object has been created in the Object Manager.

Select the Doodle object in the Object Manager. In the Attribute Manager, switch to the Object tab.



Doodle Object [Doodle Object::Perspective]

Object Properties

Current View: Perspective

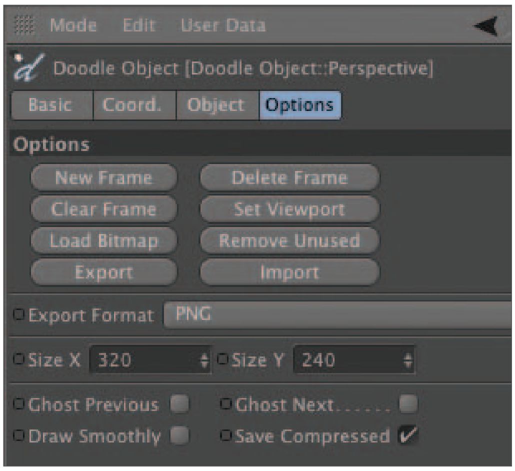
Title. . . . . Doodle Image

Description

The Attribute Manager now shows a description field. Since it is not always easy to paint legible text using a mouse or a graphics tablet, this description field can be used to type a description. This field can be dragged and dropped into the Viewport to display a typed comment on-screen, if needed. You can also double-click the HUD element itself to activate it and modify its contents.

Notice how the line you drew is fairly thick and the edges are slightly blurred.

In the Attribute Manager, select the Options tab and set the Size X value to 640 and the Y value to 480.



Doodle Object [Doodle Object::Perspective]

Options

Export Format: PNG

Size X: 320 Size Y: 240

Ghost Previous ☐ Ghost Next. . . . . ☐

Draw Smoothly ☐ Save Compressed ☑

Notice how the line is now half the width it was previously and the edges are sharper. The higher the size settings the more detail you can draw into the viewport. You can also change many aspects of the brush in the Attribute Manager.

In the Attribute Manager, select Tool from the Mode menu.

This will bring up the color options for the Doodle Paint tool. Experiment with the various settings to see how you can customize the brush. Although you can undo entire doodle strokes using the Undo command, it is also possible to erase parts of a Doodle.

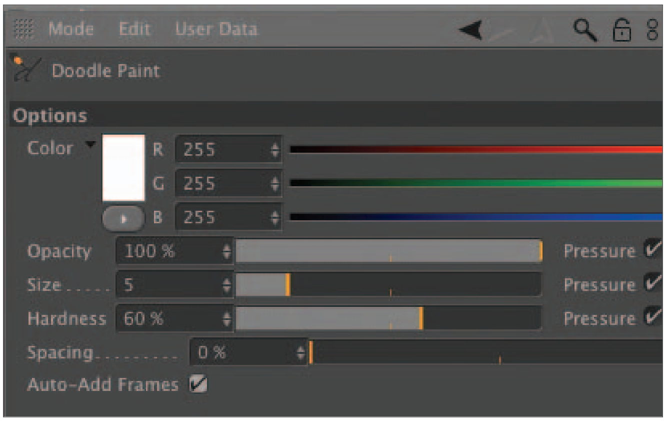
From the CINEMA 4D Tools menu, chose the Doodle Eraser

Take a look at the new options in the Attribute Manager and experiment a little more in the viewport. As you can see, Doodle painting can be quite a creative experience.

The Doodle tool can also be used to create discrete Doodle frames at different points on a Timeline.

Move the Timeline to frame 15. In the Doodle Paint tool attributes, enable the Auto-Add Frames option and paint in the viewport.

You will now see a new Doodle in the viewport completely replacing the previous image, and a keyframe has been created at frame 15 for the Doodle Object. Each time you move to a new frame and doodle, a Doodle key and a new Doodle will automatically be created.



Doodle Paint

Options

Color R 255
G 255
B 255

Opacity 100 % Pressure ✓
Size 5 Pressure ✓
Hardness 60 % Pressure ✓
Spacing 0 %
Auto-Add Frames ☑

Color

Overview

Color is everywhere, so much so that we take it for granted. Undoubtedly, if you are like most people, you assume that having color is a natural property of the objects around us. As we proceed through the following discussion you will probably be surprised to find out that what we perceive as color is perceived as such by our eyes and brains as opposed to the objects themselves. The following study of color will get rather complicated at times, but it is important to have a working understanding of the principles since color is such an important component of how objects appear to us. This background will also help you more thoroughly understand different color systems used in graphics and how to more easily use them to your advantage.

The physics of color

Computers can display colors and your eye can distinguish millions of colors. This phenomenal range is mostly the result of biology which we will address later. From a purely physical approach, the colors are limited to those of the rainbow.

Color is a characteristic of light. One way of looking at it is that light can be considered to be a wave. All waves possess the same basic characteristics. They all have a repeating pattern of crests (high points) and troughs (low points) which move along a straight line outward from the source that created them. The distance between one point on the wave and the next point along the wave where the pattern begins to repeat, is known as a wavelength.

Light has a very large range of wavelengths. Extremely long wavelengths are known as radio waves. Extremely short wavelength light is called gamma radiation. Below is a chart showing the different types of light over the range of wavelengths found in nature.

The light spectrum

long • (low f) ——————————————————————————— short •(high f)

radio -microwaves infrared -visible -UV -X rays -gamma rays

You will notice that visible light falls right in the middle of the spectrum. This might be surprising but the visible light we see is what we use to broadcast radio and TV, cook potatoes in the microwave and make x-ray images.

Visible light only makes up a very small part of the entire spectrum. The first wavelength visible to us as we move from left to right across the spectrum is about 7.5 X 10-7 m (about 30 millionths of an inch). When light of this wavelength hits our eyes, we see the color red. At the other end of the visible spectrum is a wavelength of 4.0 X 10-7 m (about 18 millionths of an inch). This wavelength of light looks blue to us. In-between are all the other colors of the rainbow in the order you probably learned them in grammar school.

What are we seeing when we see a pure color like red, orange, yellow, green or blue? We are seeing a certain wavelength of light. Another wavelength will cause us to see it as a different color.

We have all seen rainbows. While they certainly are colorful, they seem to show only a fraction of the colors you have see on TV, on a computer monitor or in a photograph. This is because these colors (e.g. mauve, olive drab, dusty rose and whatever the latest designer colors may be) have all been mixed.

The biology of color

First of all, why do we only see that particular small portion of the spectrum? Think about the source of light on earth as animals and plants evolved. The sun sends out light in all wavelengths but the wavelengths make up the biggest proportion are those of visible light. Plants use the visible portion of the spectrum from which to thrive. Some animals can see a bit more the spectrum that is visible to us. Bees and many other insects can see the lower end of ultraviolet; many flowers that look uniformly white have dramatic patterns when photographed with film sensitive to ultraviolet light.

Now, why do we see more colors than are in the rainbow when looking at the rest of the world? Why does your TV show more colors than the rainbow? Actually, the answer to both questions turns out to be the same: Color by addition. Your eye has three types of color receptors. One type detects red, another detects green and the third detects blue. Each of these receptors sends a message to the brain which in turn evaluates the message for the perception of color. All the colors you see are combinations of various amounts of red, green and blue.

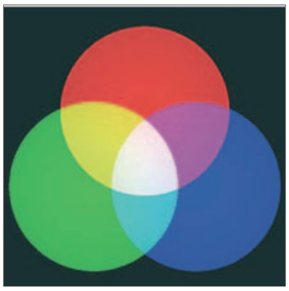
Your television or computer monitor works in a very similar way. It emits various amounts of red, green and blue light which then stimulate your red, green and blue receptors to varying degrees. So, while a rainbow produces the entire "spectrum" of colors, each of these colors is only a single wavelength. Your eye and brain, on the other hand, can mix three different wavelengths to produce a much wider array of colors.

Additive color, the RGB system

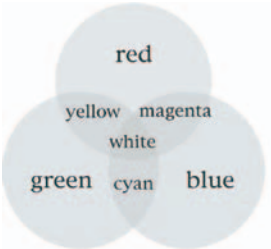
If you were asked what happens when you mix blue and yellow, you would probably say green. This is true if your are referring to paint. However, if you are mixing blue and yellow light the answer is white! In the former, the colors are subtracting from the light that reaches your eye (more about that later). In the latter, which actually turns out to be more important to us for most purpose outside of printing and photography, the colors are added. If one person shines a flashlight beam through a red filter onto a white wall and another person shines a light through a green filter onto the same spot, the resulting color will be yellow. You are perceiving red green light, but no blue light. The resulting color perceived by your brain is yellow.

All the colors you perceive can be formed by various strengths of the three colors red, green and blue. Therefore, these three colors are called the primary colors of the additive color system. To produce red, mix 100% red with 0% green and 0% blue. To produce yellow, you mix 100% red with 100% green and 0% blue. 100% of each color, will result in white.

Maybe You are getting a bit lost here. So, what is white? What wavelength produces the "color" white? None. Your brain produces the sensation of white whenever equal amounts of red, green and blue wavelengths strike your eyes at the same time in sufficient quantities.



As we mentioned before, yellow is a combination of red and green light. If you then add blue you will have all three primary colors together, resulting in white. Another way of looking at this is that subtracting blue from white light leaves you with yellow. Yellow is said to be complementary to blue. Likewise, when you mix equal amounts of red and blue you get the color magenta, which is complementary to green. Green and magenta light make white. And if you mix equal amounts of green and blue, you get cyan which is complementary to red. Or, cyan is the color remaining when you remove the red from white light.



So far, we have explained seven colors including white. Where do the other millions of colors come from? We have been talking about using the maximum intensity of the three primary colors but what if we use intensities of less than 100%. If we used 50% red, 50% green and 50% blue, there would not be enough overall intensity to produce white. They would add to produce a medium shade of gray. Ten percent of each color would produce a very dark shade of gray, almost black. As long as the three values are equal in intensity a shade of gray will result.

Now let's vary one of the colors, say 50% red, 50% green and 60% blue. The result will be a medium shade of gray with a blue tint to it. If we increase the amount of blue to 80%, we would probable refer to the resulting color as blue-gray or a washed out shade of blue. If we used 50% red, 60% green and 60% blue we would again get a medium gray only with a cyan tint. By mixing various percentages of the three primary colors we can produce all those millions of colors your computer graphics card says it can display.

Your eyes perceive color by sensing the red, green and blue light striking the receptors and your brain adding them together. It should be easy to see why the RGB color system is also used by computer monitors, television and in cinematography.

Subtractive color, the CYMK system

Let's take just a brief look at the other important color system which is used for printing. Photographs and other color printed media do not emit light to your eyes. Instead, light is reflected off a surface. When this happens, some of the colors are absorbed by the dyes or pigments in the print, leaving others to be reflected. The colors you see are those that result from certain colors being subtracted from white light.

The result is that when colors are mixed (e.g. paint) more of the light will be absorbed and will not reach the eye. This is the opposite from the additive system in which additional light is received by the eye and added to comprise a color. This opposite effect is a result of the of primary colors for the subtractive system: Cyan, yellow and magenta. So what does the K stand for?

If you have ever mixed paint or colored inks you may have tried to mix all the colors together to make black. You were probably disappointed that the best you could do was maybe a dark shade of brown. So the printing process uses black (K) in addition to the three subtractive primary colors to produce dark tones and shadows.

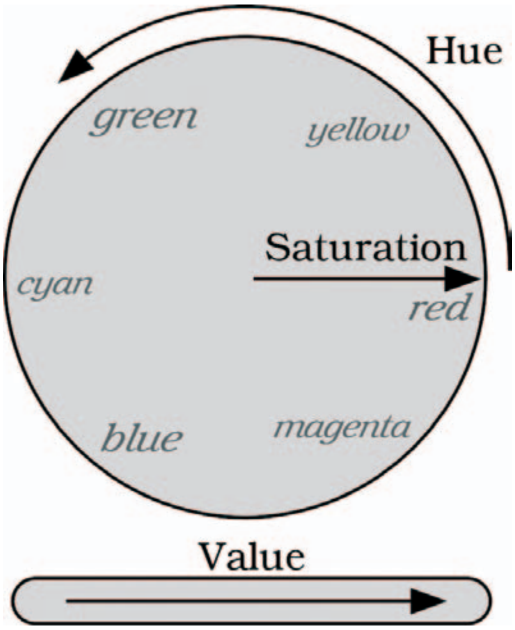
If you look at a color print in a book or magazine, you will see the photo is made up of tiny dots printed in a grid-like pattern. You will see the dots themselves only consist of CYMK colors. Other colors are created by combining different sizes of dots in each of these colors and printing them side-by-side to give the impression that they are mixed in different proportions in different areas of the photo.

Color pickers

We will conclude with a look at how colors are chosen in graphics programs. The portion of CINEMA 4D with which you specify and adjust the color is called a color picker. There are many color pickers available, but we will use just two.

The simplest color picker is the RGB color picker. It has three sliders that let you to choose how much of each primary color to blend in order to produce your final color. You can also enter numerical values for each of the RGB colors. Depending on the picker, the values may be in percentages or on a number scale from 0 to 256 (which refers to how computers work with binary numbers).

The most widely used color picker for RGB work is the HSV picker. This stands for Hue, Saturation and Value. Hue is the color as seen going around a color wheel. 0° at the outside of the wheel is red. 120° counterclockwise around the outside wheel is the color green. Another 120° counterclockwise (240°) brings you to blue. At any point directly opposite any of these primary colors you will find its complementary color.



Saturation refers to how intense the overall color is as compared to a pastel shade or one that is bleached by the addition of white. The more white that is blended in with a color, the less *saturated* that color is. On the HSV color picker the less saturated the color the closer to the center of the color wheel you are. The colors are 100% saturated at the edge of the circle.

The final color parameter in the HSV system is value. Sometimes this is referred to as brightness (the HSB system) and indicates how bright the overall color is compared to black. No matter what the other settings are, a value equal to zero always produces black. A value of 100% will give the brightest shade of whatever color has been set by the hue and saturation settings. As you decrease the value, the color will become darker.

You can use the HSV color picker in one of two ways. You can set the H and S values by clicking inside the color wheel and setting the V value using the slider. You may also set each of the three parameters by entering numerical values from 0 to 100%.

This has been a summary of some of the principles behind color. This should be enough to serve as a background to help you feel comfortable creating and using color when texturing your models. However, it should also inspire you to analyze the color of the objects around you. It helps to be able to look at a color that most people would simply call purple and recognize the proportions of red, blue and maybe even a hint of green that make up that color.

Later on we will look at how light interacts with surfaces and with transparent objects. This will help us better understand reflection and refraction, i.e. the bending of light as it passes through transparent objects.

Nature of Surfaces

Overview

Most of our perception of three-dimensional objects is a result of the interaction of two different sets of variables. First there is the shape of the object as defined by its silhouette and the shading caused by the lights cast on its surface. Second, there is the nature of the surface itself. Different surfaces create radically different appearances, depending upon a number of variables, such as smoothness, transparency, reflectivity and color.

Next to color, one of the most defining characteristics of a surface is probably its reflectivity. This can range from the shininess of most metals to the dull look of a piece of paper. A special case of reflectivity is when objects are actually smooth and shiny enough to reflect other objects.

Transparent surfaces are special in the sense that they make an object's volume apparent and let light pass into the object and interact in various ways. Refraction, or the bending of light when passing into another medium, is one of the most important of these interactions.

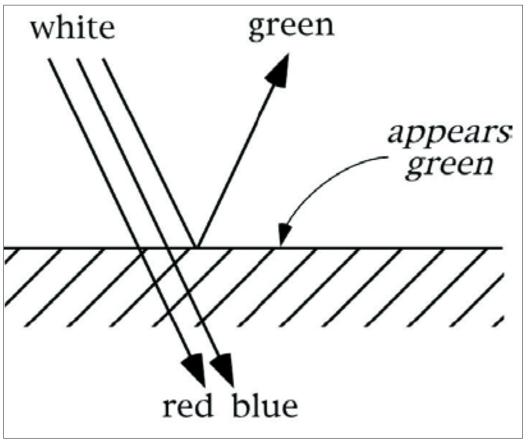
Color

Color is probably the first feature we notice about a surface. You have already read about color being a result of a mixture of different wavelengths of light striking the eye. So why are surfaces colored the way they are?

All matter is composed of atoms. The most important feature of atoms are their electrons. These electrons not only define the outer boundary of the atom but cause atoms to bond to form molecules that in turn form the objects we see. When light strikes these atoms it interacts with the electrons. Some wavelengths of light are absorbed by the atoms and then emitted as infrared heat (another form of light). Other wavelengths do not interact with the atoms' electrons and are reflected from the surface.

Different molecules have the electrons in their atoms "tuned" differently. That is, different molecules interact differently with different wavelengths of light. Some absorb blue and green light but reflect red light. Those reflected are visible to us. Other molecules may absorb red and green and thus appear blue. Molecules that reflect colors to our eyes are also called pigments.

Many surfaces contain multiple pigments. These reflect color to your eyes through the subtractive method of color. For example, if one pigment absorbed blue (reflecting red and green) and a second pigment absorbed red (reflecting blue and green) then only green light should be reflected back to your eyes. If there were more of one pigment than the other, or if one of the pigments absorbed in a slightly different region of the spectrum, the reflected color would be a different shade of green.



Some molecules have electrons that are tuned to wavelengths outside the visible spectrum. Surfaces containing these would appear white because the entire visible portion of the spectrum is reflected. However, other molecules, or combinations of molecules, may be tuned to interact with, and therefore absorb, the entire visible portion of the light spectrum. These surfaces appear black. This also explains why black surfaces get hotter upon exposure to sunlight. The energy in the absorbed visible wavelengths is re-emitted as infrared light, which we sense as heat.

Lighting and color

The above assumes that the light striking the surface is white and that it contains the entire visible spectrum. However, what if we shine colored light on a colored surface?

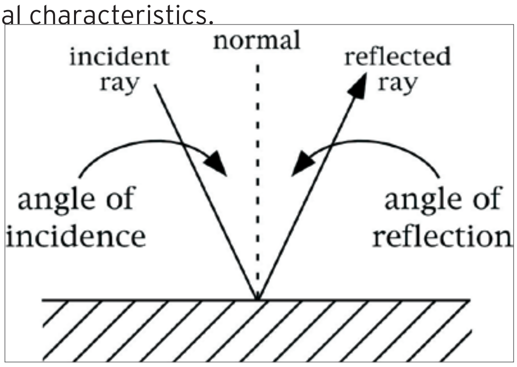
For starters, let's shine red light on a black surface. Since the black absorbs all wavelengths of visible light, it will absorb the red and the surface will appear black. No surprise here. Now let's shine red light on a white surface. Since the white surface reflects all visible wavelengths, it will reflect the red and therefore appear red. If we shine that red light onto a red surface it will also appear red.

But what if we shine the red light onto a blue (in white light) surface? Remember, the surface is blue because is absorbs red and green wavelengths. Therefore, it will absorb our red light and appear black. Likewise, shinning green light onto the blue surface will still leave the surface appearing black. Only if our light contains some amount of blue will it appear to be colored at all.

What if we shine cyan light onto the blue surface? Cyan contains both blue and green wavelengths. The green will be absorbed and the blue will be reflected, making the surface appear blue. Here is a test to see if you have really understood this: What color would a magenta (in white light) surface appear if you illuminated it with yellow light?

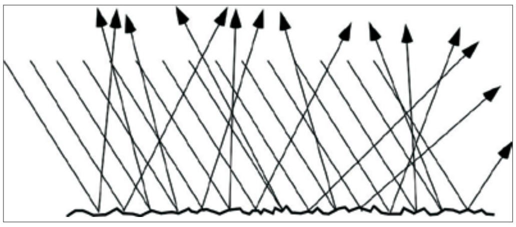
Reflection

Cinematographers divide reflection into two categories even though there are various degrees of each that blend into each other. You probably first think of a mirror when the word reflection is mentioned. This type of reflection is called **spectral reflection**. This is characterized by the formation of images of other objects on the spectral surface. Spectral reflection happens when surfaces are very smooth. Generally this happens with relatively simple substances whose atoms can be packed closely together in an orderly arrangement. This is especially characteristic of metals, although glass and certain plastics as well as finishes such as paint, varnish and lacquer show spectral characteristics.



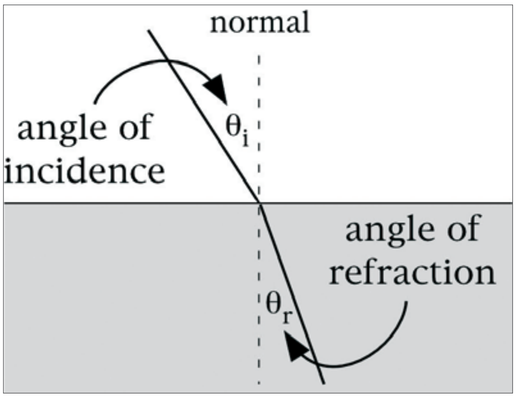
Light is reflected from a surface according to the laws of reflection. When a ray of light strikes the surface (the "incident ray") at a particular angle, it is reflected off that surface (the "reflected ray") at the same angle. Any modeling and animation software that uses ray tracing for rendering can do the calculations involved to create spectral images.

A special case of spectral reflection is found in the bright highlights that show up on surfaces as a result of lights shinning on the surface. These are called spectral highlights and, even though images of other objects are not seen, they are the reflection of the lights illuminating the surface. Smoother surfaces will generally display wider spectral highlights whereas more reflective surfaces will display brighter highlights.



The other type of reflection is termed **diffuse reflection**. All surfaces exhibit a certain amount of diffuse reflection (except the "surface" of a black hole). Diffuse reflection is a consequence of reflection from tiny areas that are aligned properly to reflect light from the light source to your eye, still according to the law of reflection. However, since the neighboring areas are not aligned properly due to the unevenness of the surface, the effect is a dull brightness in that area on the surface. In any region, again due to the unevenness of the surface, there are bound to be a few tiny areas aligned just right for some reflection to occur.

Refraction



Some substances are composed of molecules whose electrons are tuned to interact with ultraviolet light (which would be absorbed and re-emitted as infrared heat). Since visible light next to UV in the spectrum, it interacts somewhat with this sort of material. The interaction is enough to not be reflected but not enough to be absorbed. These materials are transparent to visible light and include glass, water and many plastics.

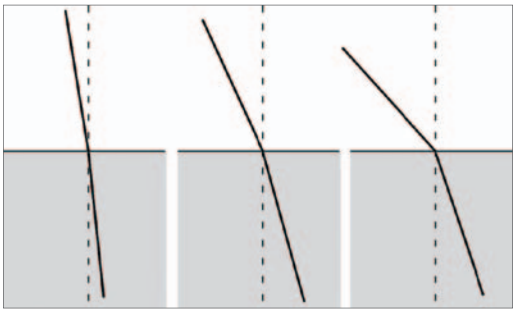
Since the visible wavelengths are interacting somewhat with the molecules of the transparent substance they get slowed down on their passage through. Because of this slowing down, rays of light entering the surface are bent from their initial path. Although we will not bother with why this bending occurs, we should more closely inspect how.

The angle of the ray striking the surface is again called the angle of incidence (•i). The angle of the ray after it enters the surface is called the angle of refraction (•r). Both of these angles are measured from the normal, a line perpendicular to the surface at the point where the ray meets the surface. When the incident ray comes in along a normal (•i = 0) the ray will pass straight through and not be refracted (•r = 0). However, as you tilt the incident ray away from the normal, the refracted ray will deviate more from the normal than the incident ray (this is assuming the light is going from a normal atmosphere into a more dense substance like glass). This is the bending of light called refraction.

If you increase the incident angle, the angle of refraction also increases. However, for this greater angle, the bending effect is exaggerated. The greater the angle of incidence, the more pronounced the angle of refraction will deviate from the angle of incidence. However, there is a simple law that helps predict what the deviation will be:

$$n = \frac{\sin\theta_i}{\sin\theta_r}$$

For a given substance such as glass, the ratio of the sine of the angle of incidence to the sine of the angle of refraction will always be a constant. The name of this constant is the **index of refraction** symbolized by the letter n.



This refraction, or bending of light, is what allows a glass lens to focus light. However, it also produces distortion effects as seen when looking through a glass of water or into a circular fish bowl. Although there are different types of glass with different optical properties, a typical value for the index of refraction of glass is about n= 1.5. For pure water the value is n= 1.33. This means that glass will produce greater distortion due to its greater index of refraction. Most software programs allow you to enter values for the index of refraction when activating a transparency channel.

Although reflection and transparency (especially with refraction) can add touches of realism to rendered images, they should be used with discretion due to the additional computing power required. You are encouraged to pay attention to the surfaces of everyday objects as you encounter them. In this way you can begin to build up a library of experience to help you when you need to model a particular surface.

Cameras

Overview

Vision is probably the most important human sense. The visual areas of the brain take up the lion's share of gray matter devoted to the senses. It is also thought that much of our memory is stored as visual information associated with these visual processing regions. This biological focus on vision has also led to the cultural emphasis on presenting information through visual means.

As a technological society It is therefore no surprise then we have developed means to record and display visual information that have become indispensable components of our current culture. Recording events and other information has probably been a part of human culture since we could first be defined as humans such as the cave paintings in France dating back 30,000 years[1]. Guttenberg's printing press was one of the most important inventions, helping disseminate information on a mass scale. However, mass production of accurate graphic images did not arrive until relatively recently with the invention of the camera and film.

In some respects, cameras have changed radically from their first use in the 1830s[2]. The basic principles have actually remained the same. This is as true of the digital cameras of today as for the cameras used to document the civil war. Although the "camera" used in a 3D software program has none of the hardware associated with a real photographic camera, most of the same principles apply in one form or another. The following is a brief survey of some important camera parameters that have relevant counterparts in the digital world.

The camera lens

The earliest "cameras" were merely pinholes leading into darkened chambers that allowed light from outside objects to pass through the pinhole and be projected onto a white screen (and later onto film). This device was called the camera obscura. Images produced in this manner are faint and lack clarity.

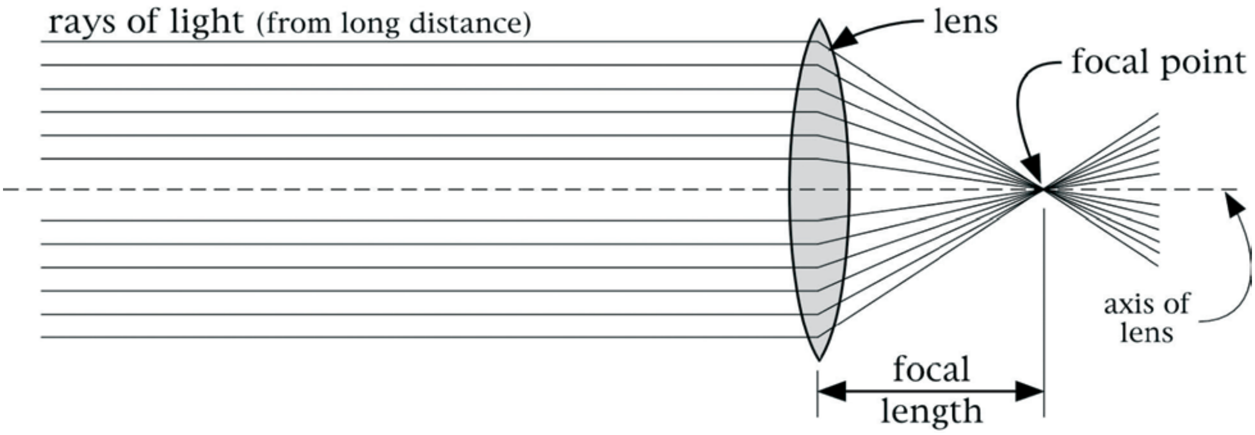
The single most important contribution to photography, besides the invention of film, is undoubtedly the lens. Lenses allow you to gather much more light than the pinhole of the camera obscura and to bring that light to a sharp focus on a piece of recording film. Although most camera lenses today are complex assemblies of several individual lenses, the principle of operation can be understood by studying a single double convex lens.

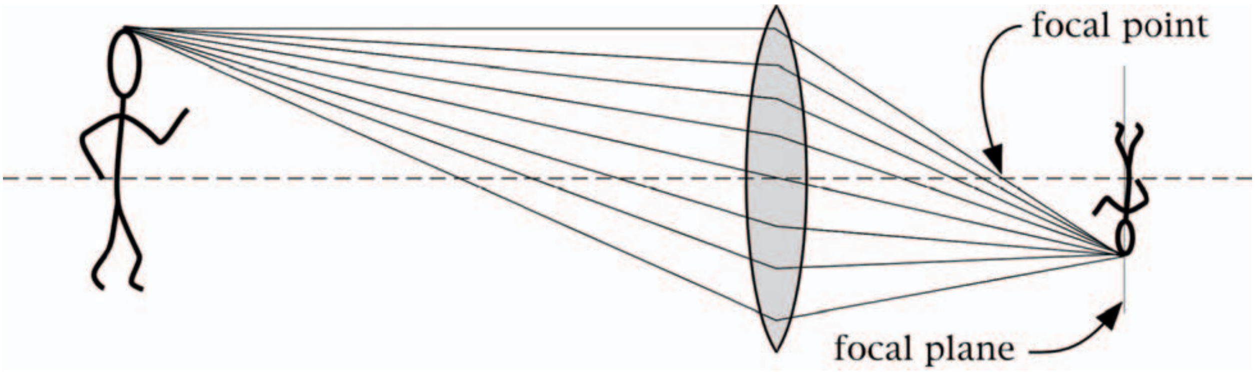
Focal length

The most important characteristic of a lens is its focal length. The types of lenses used in cameras are called converging lenses because they gather rays of light reflected from objects and bring them together (converge). This is how the lens produces a bright enough image to expose the film and makes it much more powerful than the camera obscura technique. However, to be of use, the lens must be curved in a special way.

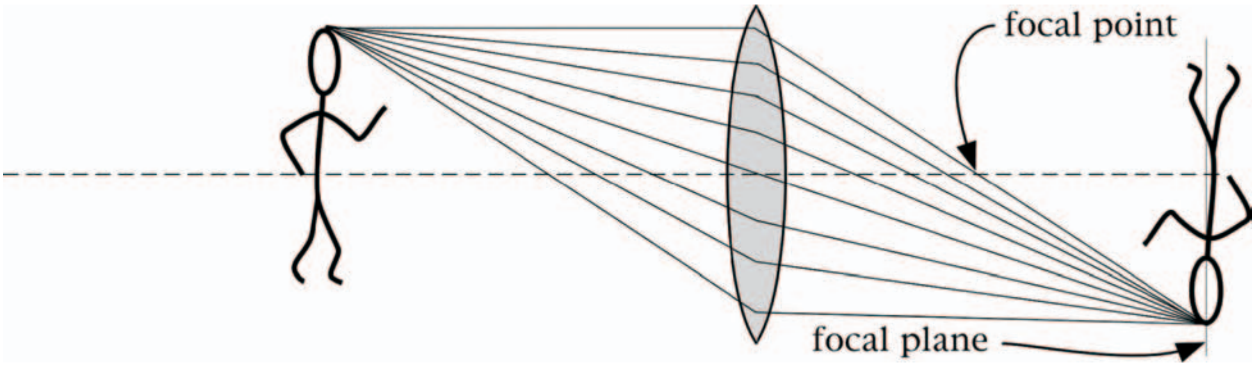
When light enters your eye or a lens from very far away, all the rays of light striking the lens are essentially parallel to each other. Lenses are ground in such a way that all these parallel rays, after traveling through the lens, all pass through the same point. This point is the focal point (focus) and the distance from the lens to the focal point is called the focal length of the lens. You are familiar with this aspect of lenses if you have ever tried to start a piece of paper on fire or fry an ant (shame on you) using a simple hand lens.

The focal length of a given lens is always constant, although zoom lenses can vary focal length due to changing the spacing between multiple lenses. So this constant focal length for a simple lens means that rays parallel to the a axis of the lens always pass through the focus. Other rays not parallel to the axis will not pass through the focus. However, here is what makes lenses useful: there is a place on the side of the lens opposite that of the object where all the rays from a certain point on the object will all be brought together (focused!).





In the above diagram, notice that all rays from the top of the figure converge at a specific point on the back side of the lens. It is at this point that an image of the top of the figure's head will be formed. Also, note that this is not the focal point. The focal point is still defined by the location at which a ray parallel to the axis (the top-most ray coming from the head in this case) is forced to cross the axis on the back side of the lens. If you were to make a similar diagram for all the rays coming from the neck of the figure, they would all come together but at a location different from where the rays emitting from the top of the head converge. This would form an image of the figure's neck at the location where the upside-down image is shown in the diagram. This image formation takes place in the same manner for each point on the figure. All these image points line up on a vertical plane to produce the total image of the figure. This is called the focal plane. This is where we would place a piece of film if we want to record the figure's image.

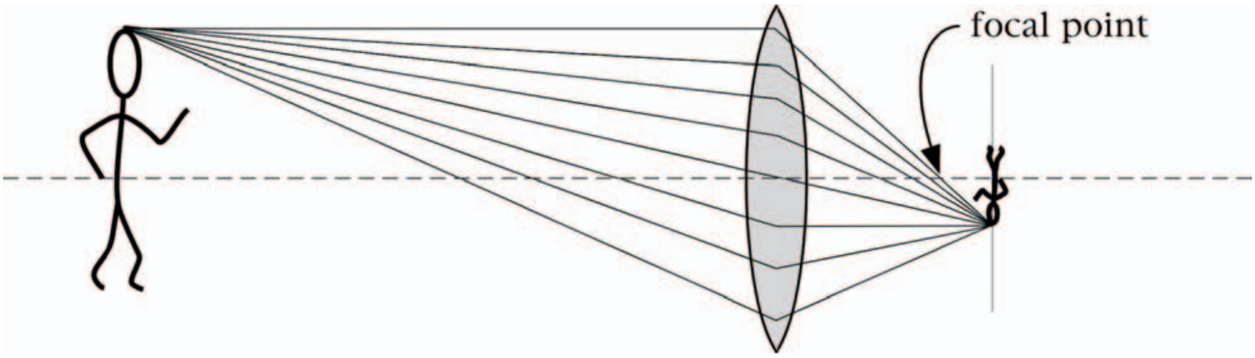


If the figure moves closer to the camera, the image will be produced in a different location. The focal plane changes (notice that the focal length, as always for a simple lens, remains the same). When you twist a lens on a camera to focus it you are changing the distance between the lens and the film so that the film is on the proper focal plane for the particular distance to the object you are focusing on. Since the film can be only on one focal plane at a time, there is only one particular camera-to-object distance where objects will be in sharp focus. The farther away an object is in front of or behind the one you are focusing on, the more out-of-focus it will be. This is called "depth of field" and will be addressed later in more detail.

One last thing you should note comparing the two diagrams is that the closer the object is to the lens, the larger its image appears. This is just as it should be. Objects closer to your eye in the real world always appear larger than they do farther away.

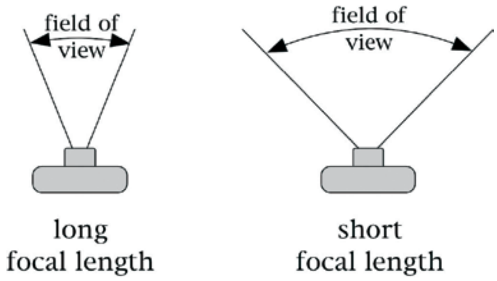
Focal length and perspective

So far our examples have used the same lens. Let's take a look at a lens with a different focal length. Compare the following diagram to the first diagram in which the figure is at the same distance from the lens.



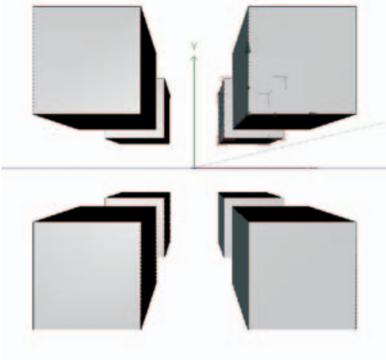
The shorter focal length results in the image being smaller when the object is at the same distance as with a lens with a longer focal length (a "longer lens"). Because the image takes up less space on the film, a wider visible range can fit onto the film. Another way of looking at this is that shorter focal length lenses have a wider angle of view - they see more of the world from peripherally and top to bottom.

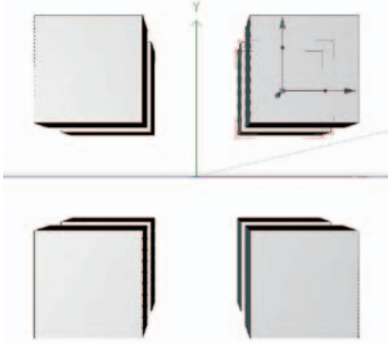
Because of this wider angle of view, short focal length lenses are called wide-angle lenses. Long focal length lenses are referred to as telephoto lenses because of the way they enlarge distant objects compared to normal lenses. The focal length of lenses is measured in millimeters (mm). A normal lens (one which has an angle of view about the same as a human eye) has a focal length of about 50 or 55 mm.



A more interesting phenomenon than just the different amount of the world taken into view happens with lenses of varying focal lengths. When a wide-angle lens is used, the difference in distance between objects close to the lens and distant objects is exaggerated. Because of the way the wide-angle (short focus) lens makes smaller images than a telephoto lens, all objects look smaller. However, with a wide-angle lens, distant objects look even smaller in comparison to closer objects. With a telephoto lens the difference in size between distant and close objects is minimized and the objects appear to not be spaced very far apart in distance from the camera. This results is a much different look between wide angle and telephoto lenses.

The two images to the right show the same eight cubes as views through two different camera focal lengths. The wide angle at the top is equivalent to a 25 mm lens, the telephoto underneath is similar to a lens with a focal length of 100 mm. The camera was moved in on the wide angle scene to make the front faces roughly the same size as those in the telephoto shot. This is not so you can compare the total field of view, but the manner in which the wide-angle lens stretches out the depth between the front and rear cubes, as compared to the telephoto lens. You will also note that the rear cubes in the telephoto shot appear almost the same size as the front cubes. However, they look substantially smaller in the wide angle shot.





F-stop and depth of field

One point which has been mentioned and can be observed from the diagrams is that lenses gather light. With a larger lens, more light will be gathered and the image will be brighter. However, larger diameter lenses have less shallow depths of field. Once you have focused on your subject not much foreground and background will be sharply focused.

However, the smaller you make the lens, the greater the depth of field. In practice, this is done with an adjustable device called a diaphragm which can be closed and opened to change the depth of field. The relative size of this opening is traditionally called f-stop. Standard ranges for a lens are from about f2.8 to f16. Lower values allow more light to enter the camera but have less depth of field. Higher values give greater depth of field but do not allow as much light to enter. This must be compensated for by longer exposure times. Since less light is reaching the film per unit of time you must allow more time for enough light to properly expose the film.

You may have noticed that the same rules that govern human sight also apply to the lens of a camera. Its easier to focus when the light is bright because your eyes' pupils become smaller to prevent overexposing the pigments of your eyes, thereby increasing the depth of field. If you have not experienced this, try threading a needle in dim candlelight then try it in a well-lit room.

Relevance to 3D modeling and animation

Even though there are no real cameras with optical lenses in your computer, CINEMA 4D must still render a scene from a given point of view, including a proper sense of depth. 3D software also uses cameras, although more in the form of mathematical processes. All the characteristics described for real cameras and lenses above have their equivalents in the 3D computer world.

Focal length is the most important of the camera parameters. You can assign different focal lengths to a camera to obtain the look you want in terms of perspective. Just as in the real world, a close up on a face with a 25 mm wide-angle lens will exaggerate the size of the nose. Likewise, a telephoto shot of a forest will make the trees look scrunched up from front to back and the forest will lack depth. The choice of focal length is also important in motion as well. When something moves from back to front in a wide angle view its size will be exaggerated enormously, appearing to accelerate a bit at the end - a good choice for spacecraft shots. On the other hand, a telephoto lens can make motion toward the camera appear to drag on, like the famous shot in *The Graduate* in which Dustin Hoffman is driving across a long bridge attempting to reach a wedding.

One special feature of computer lenses as opposed to camera lenses is that you can set an infinite focal length. This is called the "parallel" view. Using our example with the eight cubes above, the rear cubes would appear the same size as the front cubes. There would be no illusion of depth. In fact, when viewed from the front, they would be completely hidden by the front cubes. Although you would rarely use this for a render, it is indispensable in helping align objects, especially in technical projects.

The concept of f-stop has a bit of a different implication for computer applications. Since the camera does not work on the basis of light striking film, you do not need to set the f-stop to allow the proper amount of light into the camera. This is all controlled by the settings of the light objects themselves. However, remember there is another effect of f-stop - one that controls the depth of field. CINEMA 4D also offers this type of f-stop as well. You can define the depth of sharpest focus and define where in the foreground and in the background the focus should become soft.

Remember these camera and lens characteristics. Experiment with different focal lengths. This parameter can radically alter the mood of a scene, especially one with many objects at different depths. Set up multiple cameras for a scene and choose the one which gives you the desired look. However, do try to think about the look you want beforehand, a lot of layout decisions are based on the camera's focal length.