1 Object Notes

1 OBJECT ICONS

1.1 PARENTING

DIFFERENCES FROM OTHER OP TYPES

As opposed to other types of OP tiles, Object tiles you place in the *Object* Layout area do not need to be connected to one another. They simply specify the existence of an object in the scene.

If you connect the output of one object to the input of another, you create a parenting structure where the objects become linked in a hierarchical relationship. Parenting defines the relationship between objects that you want to treat as a linked group.

An object which is parented to another object inherits the transformations of the parent object. This means that moving the parent moves the child as well. Each object can have only one parent. In addition, other information is inherited from parent objects. For example, motion blur can be inherited from a parent object. If the rendering parameters are not enabled for an object, they will be inherited from the parent object.

If, for example, you wanted to include as part of your scene, the perspective on the world as seen from a particular object, you could link the output of a geometry object to the input of a camera object. Defining a parenting relationship in this way means that any transformations you make to the geometry will move the camera along with it.

1.2 PARENTING IN THE VIEWPORT

If you don't want to manually wire up the connections between Object tiles in the Network Editor, you can also perform parenting directly in the Viewport with the Parent Operation – see *Parent Operation* p. 341.

2 Objects

1 COMMON OBJECT PARAMETERS

1.1 PARAMETERS – TRANSFORM PAGE

KEEP POSITION WHEN PARENTING

This parameter is valid for the: Fog, Bone, Camera, Force, Geometry, Light, Microphone, Null, and Sound objects.

If this option is enabled, then when the object is reparented, its current world position will be maintained. This is done by changing the objects transform parameters.

TRANSFORM ORDER

The menu attached to this parameter allows you to specify the order in which the changes to your object will take place. Changing the Transform order will change where things go much the same way as going a block and turning east gets you to a different place than turning east and then going a block.

ROTATE ORDER

The rotational matrix presented when you click on this option allows you to set the transform order for the object's rotations. As with transform order (above), changing the order in which the object's rotations take place will alter the object's final position.

TRANSLATE / ROTATION / SCALE

The three fields allow you to specify the amount of *Translation* along any of the three axes; the amount, in degrees, of *Rotation* around any of the three axes; and a non-uniform *Scaling* along the three axes.

As an alternative to entering the values directly into these fields, you can modify the values by manipulating the object in the Viewport with the Transform Operation.

Transforms for objects are stored using double precision floating point values, so calculations for positioning objects is very accurate.

Tip: To take an existing view you've arrived at in the Viewport by using the manual controls (dolly, zoom, tumble, etc.) and get the same view for the camera, use the Camera ✣ parameters, and click on the *Save View to Camera* button. One caveat – if the orientation of the camera is being determined by using a Rotation Handle or an object's *Look At* parameter, the view may not be properly saved.

camera shake

To make the camera appear to shake during a render, try these expressions in the camera's Translate-Z parameter:

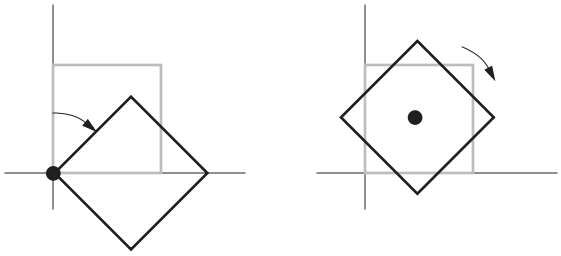
Shake on 2s:
if($F%2, 1, -1) * easeout()

Shake on 4s:
if($F/2%2, if($F%2, 1, 0.5), if($F%2, -1, -0.5)) * easeout()

PIVOT

The Pivot point edit fields allows you to define the point about which an object scales and rotates. Altering the pivot point of an object produces different results depending on the transformation performed on the object.

For example, during a scaling operation, if the pivot point of an object is located at -1, -1, 0 and you wanted to scale the object by 0.5 (reduce its size by 50%) the object would scale toward the pivot point and appear to slide down and to the left.



Pivot at 0, 0, 0
Rotate 45° in Z

Pivot at 0.5, 0.5, 0
Rotate 45° in Z

In the example above, rotations performed on an object with different pivot points produce very different results.

UNIFORM SCALE

This field allows you to change the size of an object uniformly along the three axes.

Note: Scaling a camera's channels is not generally recommended. However, should you decide to do so, the rendered output will match the Viewport as closely as possible when scales are involved.

LOOK AT

Allows you to orient your object by naming the object you would like it to look at, or point to. Once you have designated this object to look at, it will continue to face that object, even if you move it. This is useful if, for instance, you want a camera to follow another object's movements. The *Look At* parameter points the object in question at the other object's origin.

LOOK AT UP VECTOR

When specifying a lookat, it is possible to specify an up vector for the lookat. Without using an up vector, it is possible to get poor animation when the lookat object passes through the Y axis of the target object.

Don't Use Up Vector — Use this option if the look at object does not pass through the Y axis of the target object.

Use Up Vector — This precisely defines the rotates on the object doing the looking. The up vector specified should not be parallel to the look at direction. See *Orient Up Vector* p. 259 below.

Use Quaternions — Quaternions are a mathematical representation of a 3D rotation. This method finds the most efficient means of moving from one point to another on a sphere.

PATH OBJECT

Names the object that functions as the path you want this object to move along. For instance, you can name an object that provides a spline path for the camera to follow.

tip – for smooth motion along a path

Having an object follow an animation path is simple. However, when using a NURBS curve as your path, you might notice that the object speeds up and slows down unexpectedly as it travels along the path. This is usually because the CVs are spaced unevenly. In such a case, use the Resample SOP to redistribute the CVs so that they are evenly spaced along the curve. A caution however – using a Resample SOP can be slow if you have an animating path curve.

An alternative method is to append a Basis SOP to the path curve and change it to a *Uniform Curve*. This way, your object will move uniformly down the curve, and there is no need for the Resample SOP and the unnecessary points it generates.

ROLL

Using the angle control or the editing filed, you can specify an object's rotation as it animates along the path.

POSITION

This parameter lets you specify the position of the object along the path. The values you can enter for this parameter range from zero to one, where zero equals the starting point and one equals the end point of the path. The value slider allows for values as high as ten for multiple "passes" along the path.

ORIENT ALONG PATH

If this option is selected, the object will be oriented along the path.
The positive Z axis of the object will be pointing down the path.

ORIENT UP VECTOR

When orienting an object, the up vector is used to determine where the positive Y axis points.

AUTO-BANK FACTOR

The *Auto-Bank* factor rolls the object based on the curvature of the path at its current position. To turn off auto-banking, set the bank scale to 0.

1.2 PARAMETERS – MISC PAGE

USE DISPLAY COLOR

When this parameter is enabled, and the object is being displayed in wireframe mode in the Viewport, the object will be displayed using the colour specified below.

DISPLAY COLOR

Set the desired display colour by specifying an RGB value here, or by using the HSV sliders. Clicking the ✣ button displays a colour dialog that allows you to select from a range of predefined colours. Click on the desired color's name, or on the color swatch, to select that color. The name of the selected color appears highlight to indicate that it is active.

hue / saturation / value

This mode of color definition allows you to specify the Hue, the Saturation and the Value of a given color. Hue describes a color's place on the spectrum. Saturation defines the amount of white mixed in with the color, while Value describes the lightness or darkness of the color. You can edit the values by dragging the slider or by manually entering the values in the edit fields.

red / green / blue

This mode of color definition allows you to specify the Red, Green, and Blue values that are mixed together to form the color you want. You can edit the amounts by dragging the slider, or by manually entering the values in the edit fields.

VIEWPORT SELECTING ENABLED

Enabling this button changes the object's status to *selectable*, meaning that the object is capable of being selected for specific commands and of being singled-out as an editable object in the Viewport.

SELECTION SCRIPT

Clicking the button to the right of the edit field displays an open file dialog that lets you search for a script to execute when you click on the selected object in the Viewport using the Select Operation.

2 AMBIENT LIGHT OBJECT

2.1 DESCRIPTION

The Ambient light Object controls the color and intensity of the environmental light in a given scene. This light, unlike the Light Object, has no particular source. The light it sheds comes from everywhere as opposed to a point light source or focused spotlight.

2.2 PARAMETERS – LIGHT PAGE

LIGHT COLOR

You can modify the colour of the light three ways: Colour List, Hue, Saturation, and Value, or Red, Green, and Blue. To choose one, click on the appropriate box and the color editing fields below change accordingly.

color list

Clicking on the dialog button (✣) to the left of the HSV and RGB buttons invokes a dialog that allows you to select from a pre-defined list of SGI colours. To select a color, use the scroll bar to view the available colors, and click on either the sample color swatch or on the name of the colour.

hue / saturation / value

This mode of color definition allows you to specify the Hue, the Saturation and the Value of a given color. Hue describes a colour's place in the spectrum. Saturation defines the amount of white mixed with the colour, while Value describes the lightness or darkness of the colour.

red / green / blue

This mode of colour definition allows you to specify the Red, Green, and Blue values that are mixed together to form the colour you want.

DIMMER

This parameter allows you to change the intensity of the light either as a static value or over time.

2.3 PARAMETERS – RENDER PAGE

The *Render* parameters control the object's behaviour and appearance when rendered. Only additions to the standard parameters are listed below – for detailed discussion of the rest of the parameters, please see *Parameters – Render Page* p. 283.

2.4 PARAMETERS – MISC PAGE

For a complete description, see *Parameters – Misc Page* p. 259.

DISPLAY COLOR

The colour to use when displaying the object in the Viewport.

VIEWPORT SELECTING ENABLED

Allows the geometry to be selected in the Viewport

SELECT SCRIPT

The script to run when object is picked.

3 ATMOSPHERE (FOG) OBJECT

3.1 DESCRIPTION

The Atmosphere Object affects the lighting in the render such that the atmosphere appears filled with fog. Creating atmospheric effects in Houdini is straightforward when you realise that there are two objects working together to create the effect:

- the Light object
- the Background object

The light object creates the visible cone and the background object is required for all visible light rays to strike.

Atmospheric lights also project through the field of view of the light source so that both depth shadowing and projection maps work.

When creating atmospheric lights, cone angle is ignored. The best thing to do to get a cone light is to place a projection map of a circle in front of the light (try a Radial-Concentric ramp in the COP Editor). To get a nice effect, you may wish to fractalise the picture slightly or create noise in the picture. This will give you more streaks in the atmosphere and make it look a little more realistic.

The depth shadows will allow objects to cast shadows through the fog.

It is important to turn on Zoom and Focal length in the Atmosphere object. Check the field of view by looking through the light source. Whatever you see through the light source will be filled with volumetric fog when you render.

Set the *Attenuation* in the Atmospheric dialog. This gives you the ability to have the atmosphere fall off as it would in real life.

Note: Mantra automatically adds a background object that encompasses the scene in order to make the atmosphere visible. The object is simply a large box with a matte shader applied to it. You can disable this behaviour by setting the environment variable RAY_NO_FOGBOX.

3.2 PARAMETERS – TRANSFORM PAGE

The transform parameters alter the translation, rotation, and scale of the Atmosphere object. Some atmosphere shaders (i.e. Layered Fog) may require transformations. For a more detailed discussion of these parameters, please refer to the discussion *Translate / Rotation / Scale* p. 256.

3.3 PARAMETERS – SHADING PAGE

SHADER

Clicking on the ✣ button to the right of the edit field invokes a dialog script that allows you to set the parameters for the Shader type. These are described in detail in *Atmosphere Types Dialog* p. 352.

RMAN SHADER

Clicking on the ✣ button to the right of the edit field invokes a dialog script that allows you to set the parameters for the Shader type. Select from the predefined scripts *depthcue* and *fog*.

SHOP VOLUME SHADER

Click the ✣ button to invoke a dialog script to set the SHOP Volume Shader.

3.4 MISC(ELLANEOUS) PAGE

Sets the display colours and picking. For a discussion of this tabbed page's parameters, please refer to *Parameters – Misc Page* p. 261.

3.5 EXAMPLE

1. Change the Font SOP in the *logo* object to "H"; change its *Material* to: *blue_plastic* .

2. Add an Atmosphere object to a default Houdini environment.

3. Move light1 to be behind the "H" object. Say something like *Translate:* 0, 0, -1.5 ;
and change its color to green.

4. Increase the Ambient light a bit. Say *Value:* 0.3.

5. View through light1 and adjust the Viewport controls so that you see the whole of the text. If you like, adjust the *Resolution* of the light to be smaller for faster Z-depth rendering.

6. Turn on *Auto-Generate Depth Map* (*Shading* page) in light1;
Set the *Z-Depth Map* file to be: */tmp/$OS.pic* ;

7. Set the *Projection Map* for light1 to be: *$HH/pic/circle.pic* ;

8. Open up the Atmosphere ✣ dialog for light1:

- change *Density* to: 1
- change *Attenuation* to: 2
- turn on *Projection Map*
- Change *Sampling Quality* to: 0.1 (faster rendering)
- Turn on *Focal/Aperture*

9. Change view back to *cam1* using the Camera menu (*View* Operation).

10. Render the scene by selecting *mantra1* from the Render icon's pop-up menu.

4 BLEND OBJECT

4.1 DESCRIPTION

The Blend object allows various effects such as blended inputs, animating the parents of objects, sequencing, partial transformation inheritance, three-point orientation, and other effects. It gives you some extra flexibility in setting up parent-child relationships. It operates like the Switch and Sequence blend SOPs insofar as it takes more than one input and blends or switches those into one output.

Some potential uses of the Blend OP are to animate parenting, such as when one character passes an item to another, or to pass on only part of a parents characteristics.

The Blend Object gives you some extra flexibility in setting up parent-child relationships. It operates like the Switch and Sequence blend SOPs insofar as it takes more than one input and blends or switches those into one output. As with the other objects, such as lights and cameras, the Blend OP can be placed and wired into networks, but its only purpose is to channel data rather than hold it.

Blend allows for a great deal of control over how much influence each of the inputs has over the child object. This can be set either using a weight value for the input or by masking the input to restrict its effect by channels. Each input has its own set of weight and mask controls. There is a special case when you have exactly three inputs which enables the *Normal Offset* and *Orient Axes* parameters. The Offset works by moving the child perpendicular to the plane formed by the three inputs; the *Orient* function rotates the child such that the first input determines the axes centre, the second the +X orientation, and the third the +Y orientation.

The *Sequence* type simply switches between inputs, so the weight and mask controls become inactive. A *Sequence* value of 0 parents the child to the first input object, a value of 1 parents it to the next, etc. Any value outside the range of inputs will unparent the child.

4.2 PARAMETERS – PARENTS PAGE

TYPE */parenttype*

Method in which parent transforms (i.e. Translate, Rotate, Scale) are combined to produce a unique blended transform.

SEQUENCE */sequence*

When the type is set to *Sequence* this value controls which input parent contributes to the childs position. It can be used to animate a child from one parent to another.

A sequence outside the range of inputs is interpreted as unparenting the child.

When the type is set to *Constrain*, this is similar in operation, except that no 'popping' occurs when transferring from one parent to another.

SHORTEST PATH ROTATION BLENDING (SEQUENCE ONLY)

Will use the shortest path possible when blending rotations.

RESET FRAME (CONSTRAIN ONLY)

Since this behaviour is state-dependent, this value allows the child to reset itself (i.e. ero any accumulated transform) at the frame.

WEIGHT 1 / 2 / 3 (BLEND ONLY) */blendw1*

These weights are used to weight each corresponding input parent.

MASK 1 / 2 / 3 (BLEND ONLY) */blendm1*

These masks are used to select which component of each parent is used in the blending process.

NORMAL OFFSET (BLEND ONLY) */noffset*

When exactly three parents are input, the child position may be offset in the direction perpendicular to the triangular plane they form.

ORIENT AXES (BLEND ONLY) */axesorient*

When exactly three parents are input this option will orient the child's local axes to match the orientation of the parents as follows:

First Parent:	Axes Centre
Second Parent:	Axes +X
Third Parent:	Axes +Y

4.3 PARAMETERS – TRANSFORM / MISC

These parameters are identical to those found in other objects and allow for a secondary transformation to be applied after the initial parent blending stage. For more information on transformation parameters, see *Parameters – Transform Page* p. 267.

5 BONE OBJECT

5.1 DESCRIPTION

The Bone Object is used to create hierarchies of limb-like objects that form part of a hierarchy or chain of bone objects that are parented to one another. The movement of the chain of Bone objects is "solved" or computed based on several methods including Inverse Kinematics. The parenting attribute of bones is unique in that each bone attaches to the end, not the origin, of the parent bone.

It is recommended that you use the Bones Operation, accessed through the icons above the Viewport, to construct such a chain because placing individual bone objects and establishing their parenting relationships from operator to operator is extremely time consuming and not at all intuitive. In addition, a chain created using the Bones Operation will produce a better behaved bone chain.

By default, bones do not render. They contain two types of display geometry: "link geometry" and "capture region geometry". The former consists of a narrow diamond shape which has been stretched to the length specified in *Bone Length*, and placed along the Bone Object's negative Z-axis. The latter consists of two or more user-controllable pill-shaped regions that are used to define capture regions used in skeleton SOPs. You can specify whether either of the two types of geometry are displayed.

The actual movement of the Bone objects is controlled through an IK CHOP (when using the standard Capture/Deform model). The IK CHOP overrides the bone rotation parameters. If you want to override this behaviour, then you need to delete the bone's channels such that they are no longer over-ridden by CHOP control.

When bones are parented as children of bone objects, they are automatically positioned at the tip of their parent bone object. If a bone object is parented as the child of a non-bone object, it will attach to the origin of the parent object.

5.2 PARAMETERS – TRANSFORM PAGE

By default, the translate parameters are locked at 0 so that child bone objects do not come apart. You will need to unlock the Translate parameters if you wish to have the child bone object come apart.

The Bone object's transformation parameters are a slightly abbreviated version of those found in other Object operations. The *Transform Order* is *Scale, Rotate, Transform* and should not be changed. The rotation order is ZYX and should not be changed. The scale is 1 and should not be altered.

For more information on transformation parameters, see *Parameters – Transform Page* p. 267.

5.3 PARAMETERS – BONE PAGE

DISPLAY LINK

Enable this option to toggle the display of link geometry in the Viewport.

REST ANGLES */Rx /Ry /Rz*

Rest angles of bone chains are generally determined by their position at the capture frame (See *Bones Operation* p. 310). They define the relative weighting of rotations about the bone's X, Y, and Z axes for the IK CHOP.

BONE LENGTH */length*

This parameter changes the overall length of the bone. It affects the size of the bone geometry and the positioning of the capture geometry and also determines the "end" of the bone – i.e. where Bone Objects that are children of this bone will be positioned. By default, its end is oriented so that it lays along the local negative Z axis.

DAMPENING */ikdamp*

Affects how quickly this bone's angles can be changed. This parameter prevents bone chains from locking in a dramatically straight line when they are fully extended. This type of locking motion tends to make characters look robotic. The dampening field is a number between zero and one. The larger the number, the greater the dampening effect. Dampening is applied to the entire bone chain as the chain approaches its full extension. The end affector is thus allowed to drift off the end of the bone chain. The net effect is that when the chain is nearly fully extended, a relatively large end affector motion will cause a relatively small motion in the end of the chain. This provides finer granularity in controlling the bone chain when it is fully extended.

Dampening applies to the following solver types:
Inverse Kinematics and *Inverse Kinematics with Constraints*.

DAMPING ANGLES / RANGE / ROLLOFF

/beginxrange /beginyrange /endxrange /endyrange /xrolloff /yrolloff

These parameters specify the damping angles for the bone object. The damping angles, range, and falloff determine where and when to lessen the effect of the angular contribution of a joint in the solver in X or Y axes.

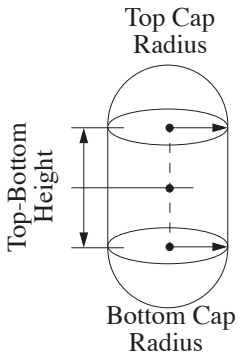
Range / Angle / Rolloff applies to the following solver type:
Inverse Kinematics with Constraints only.

5.4 PARAMETERS – CAPTURE PAGE > CAPTURE-DEFORM SOPS

The Capture page is used to automatically create a capture region used by either the Capture / Deform OPs *(see Capture OP* P. 472 *and Deform OP* P. 541 *of the Geometry section), or the Skeleton* OP *(obsolete – see Skeleton OP* P. 745*). The capture region consists of at least two 'ellipses', that may be translated along the bone's length.*

DISPLAY CAPTURE GEOMETRY

Toggles the display of the capture ellipses.



REGION CENTRE / TOP BOTTOM CAP HEIGHTS

All the geometry that is generated by the Bone object is created using regular SOPs. Therefore, it is customisable by simply changing the Bone Object's SOP network. The bone's display and render geometry can be easily altered by changing its contained SOPs.

REGION CENTER

Position of the center of the region.

TOP/BOTTOM HEIGHT */theight /bheight*

Height of the region from the centre to the top cap.

TOP/BOTTOM CAP */tcapx-z /bcapx-z*

The X, Y, Z radii of the top/bottom hemisphere.

5.5 PARAMETERS – CAPTURE PAGE > SKELETON SOP

The Skeleton SOP is generally obsoleted by the Capture and Deform SOPs. It is provided for backwards compatibility to Houdini 2.0 and earlier.

DISPLAY CAPTURE REGION

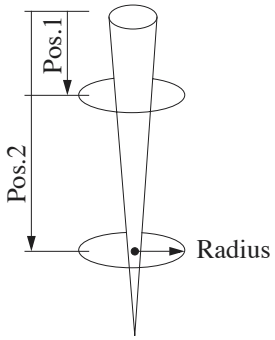
This toggle enables or disables the display of the capture regions on the bone. Enabling it activates the parameters below.

JOINT DIVISIONS */jointdivs*

This parameter is only enabled when the parent of this object is another bone. Between the two bones, a number of ellipses will be placed based on the number of divisions. These ellipses give greater control over the capture regions at the joints between bones.

JOINT SCALE /jointsx,y

Scales the joint along the axis of the bone, and away from the bone respectively.



ELLIPSE 1/2 POS /ellipse1tz /ellipse2tz

Specifies the relative distance of the first/second ellipse to the start of the bone. The default is 0.

ELLIPSE 1/2 RADIUS /ellipse1radx,y /ellipse2radx,y

The radius of the first/second capture ellipse. Note that this is radius is scaled by the length of the bone as well.

GRAB CAPTURE ANGLES

Contains a series of bone chains for which you can choose to have the Capture Angles set to the current rotate values of each bone in the specified bone chain.

CAPTURE ANGLES

These are the values that are used for the Show Capture Position solver. The capture angles can be set manually by typing in the fields or by using the *Grab Capture Angles* menu. They are also automatically set to the rotate values of the bone at the time when the *Capture Points* button of the Skeleton SOP is hit, if the bone's capture region is used in that Skeleton SOP.

5.6 PARAMETERS – SHADING PAGE

The shading parameters control the application of textures to an object in the scene. The Bone object's shading parameters are identical to the Geometry object's. For more information see *Parameters – Shading Page* p. 280.

5.7 PARAMETERS – RENDER PAGE

For a more detailed discussion of each of the parameters associated with this tab, please refer to *Parameters – Render Page* p. 283.

5.8 PARAMETERS – MISCELLANEOUS PAGE

The *Miscellaneous* parameters control variables like the display color of the geometry in the Viewport and the object's selectable status. For a more detailed discussion of the parameters associated with this tab, see *Parameters – Misc Page* p. 261.

6 CAMERA OBJECT

6.1 DESCRIPTION

Camera Objects are objects that act like real-world cameras – you view your scene through them, and render from their point of view. Cameras can be attached to geometry objects by parenting them (see *Parenting* p. 255).

Tip: To designate a centre of interest for the camera that doesn't appear in your scene, create a Null object and disable its display flag. Then parent the camera to the newly created Null object, and tell the camera to look at this object using the *Look At* parameter. You can direct the attention of the camera by moving the Null object with the Select Operation. If you want to see both the camera and the Null object, enable the Null object's display flag, and use the Select Operation in an additional Viewport.

6.2 PARAMETERS – XFORM PAGE

The *Transform* parameters alter the translation, rotation, and scale of the object. For a more detailed discussion of these parameters, please refer to the discussion *Translate / Rotation / Scale* p. 256.

6.3 PARAMETERS – VIEW PAGE

PROJECTION

A pop-up menu lets you choose from *Perspective*, *Orthographic*, *Polar Panoramic*, and *Cylindrical Panoramic* projection types. The Viewport will only display orthographic or perspective views.

ORTHO WIDTH

Only active if *Orthographic* is chosen from the *Projection* pop-up menu. This specifies the width of the orthographic projection.

RESOLUTION

Allows you to set the image's resolution according to either your own specifications or, using the pop-up menu, according to commonly accepted standards like Abekas or NTSC.

PIXEL ASPECT

This parameter sets the pixel aspect ratio of the pixels for the intended display device. Not all devices have square pixels, even though the computer monitor's display has. For example, Abekas recorders have rectangular pixels. If you intend to

record to an Abekas device in NTSC, set the aspect value to 0.889 and render your pictures to a resolution of 720 × 486. The Silicon Graphics VideoFramer will also accept these images, as well as images rendered at 645 × 486 with a pixel aspect of 1.0. For more information, see *Pixel Aspect* p. 755 in the *Outputs* section.

Note: If a picture of a circle is rendered with a pixel aspect of 0.889, the circle will appear to be oblong on the computer monitor. When recorded to an Abekas, however, the circle will be correct. this applies only to the raster image produced by the render, not the image displayed in the Viewport.

FOCAL LENGTH (IN MM)

This parameter sets the focal length of the lens, zooming in and out. Perspective is flattened or exaggerated depending on focal length. This parameter does not create curvature effects such as fish-eye lenses. For that, see the *Lens Curvature* parameter.

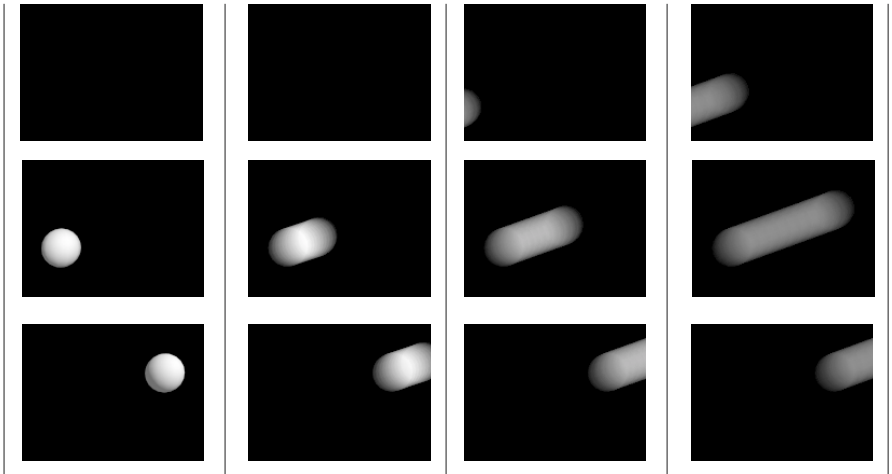
APERTURE (IN MM)

This value relates to the area through which light can pass for the camera.

SHUTTER SPEED

The portion of the frame interval (between 0 and 1) that the camera shutter is open, used to determine the amount of motion blur.

A *Shutter* value of 1 means that the whole frame is exposed to light. The term "frame" here means the time from one exposure to the next, so if you render on twos, the "frame" value is twice as long. The motion will be blurred from the beginning of the first frame to the end of the second frame. Specifying a shutter value of 0.5 in this case would blur from the beginning of the first frame to the end of the first frame. No motion from the second frame would be exposed on the film.



Without Motion Blur | Motion Blur Shutter = 0.3 | Motion Blur Shutter = 0.6 | Motion Blur Shutter = 1.0

NEAR / FAR

This control allows you to designate the near and far clipping planes. Objects closer / further away from the lens than these distances will not be rendered.

FOCUS

This parameter determines lens focal distance and sets the distance from the camera at which objects will be in focus. If the *fstop* parameter is also used, objects outside the focus distance will be blurred.

F-STOP

Lens *f-stop* determines the blurriness of depth-of field effects.

LENS CURVATURE

Creates a lens curvature distortion similar to wide angle or fish eye lenses if set to values greater than 0.

Micropolygon rendering (mantra -a) needs to be turned off in order to work.

6.4 PARAMETERS – CROP PAGE

CROP LEFT / RIGHT / BOTTOM / TOP

These parameters define the cropping area in terms of the margins of the camera's viewing area. Cropping restricts the camera's field of vision when rendering a scene allowing you to visualize only a portion of your work.

WINDOW X / Y * /winx /winy*

These parameters define the center of the window during the rendering process. The window parameter takes the area defined by the cropping command (see above) and expands it to fit the camera's field of vision. It is important to note that this action is independent of perspective. In other words, it acts as though you are zooming in with the camera without actually moving the camera.

WINDOW SIZE * /winsize*

The *Window Size* parameter specifies the dimensions for expanding the cropped area specified in the *Crop* parameter above.

WINDOW ROLL * /winroll*

This parameter sets the amount, in degrees, the window area rolls. This can be set as a static value or as an aspect that changes over the course of the animation. The roll occurs about the centre of the window.

Production Tip: The */winx-y /winsize* and */winroll* channels can be used to pan a background when rotoscoping CG imagery over a static plate. These model a traditional animation camera stand which has translation, zoom and roll properties.

6.5 PARAMETERS – ROTO PAGE

To rotoscope background images into a Viewport, use the Viewport display options (the ✣ button at the bottom-right of the Viewport) dialog. In the *Options* page, the *Background Images* parameter controls whether background images are rotoscoped into the Viewport. If enabled, it will extract the COP Network and the COP name from the Viewport's camera COP Net and COP name fields from the Rotoscope parameters. This means rotoscoping is relative to the current camera that you have selected. So if someone has filmed different shots from different cameras, you can assign those specific shots per camera.

You can override these settings in the Viewport display options, but the next time the scene is cooked, it will be set back to the camera's Rotoscope settings (Camera object > *Rotoscope* page).

COMPOSITE NETWORK

Selecting this option displays a pop-up menu that lists the networks of COPs (Composite Networks) available for use in rotoscoping.

COP NAME

This pop-up menu displays the individual COPs that make up the network you select.

COP FRAME

The *Cop Frame* parameter allows you to specify which frame(s) you want to use from a COP. By default, this is set to $F (turn *Expressions* on to change) so the frame number will change to playback all frames loaded into the COP.

Delete the channel (by clicking, and selecting from the menu), and enter a number like "1" if you want a static background.

PLANES

Specify which COP colour planes should be included.

APPLY ZOOM

If enabled, the camera window channels and zoom will affect the rotoscoped image.

6.6 PARAMETERS – RENDER PAGE

The rendering parameters associated with the Camera object are for the ability to set rendering properties once in a hierarchy of objects. You need only set the parameters once and any objects linked to this object in a parenting relation inherit these rendering parameters. For a more detailed discussion of each of the parameters associated with this tab, please refer to *Parameters – Render Page* p. 283.

6.7 PARAMETERS – OPENGL PAGE

This page contains parameters which affects the applications run-time view settings for controlling the OpenGL display; but not any rendered output.

CLEAR VIEWPORT

This channel can be used to partially clear the Viewport between redraws. By default, the Viewport is entirely cleared between each frame, but by setting this value lower than one, many interesting results can be achieved. These include real-time fades, trails and streaks.

FOG TYPE

This menu determines the type of fog rendered in the OpenGL Viewport:

Linear fog uses the following equation:

$$f = \frac{end - z}{end - start}$$

Exponential fog uses the following equation:

$$f = \frac{-density * z}{e}$$

Squared Exponential fog uses the following equation:

$$f = \frac{-(density * z)^2}{e}$$

Regardless of the fog mode, f is clamped to the range [0,1] after it is computed. Then, if GL is in RGBA color mode, the fragment's color Cr is replaced by:

$$C_r' = fC_r+(1-f)C_f$$

DENSITY

A value that specifies density or thickness, used in both exponential fog types. Only non-negative densities are accepted.

NEAR

The *near distance* used in the linear fog equation.

FAR

The *far distance* used in the linear fog equation.

COLOR

The colour of the fog.

ALPHA

Used to control the background opacity of the scene.

6.8 PARMS - MISC PAGE

For a complete description, see *Parameters – Misc Page* p. 259.

DISPLAY COLOR

The colour to use when displaying the object in the Viewport.

VIEWPORT SELECTING ENABLED

Allows the geometry to be selected in the Viewport

SELECT SCRIPT

The script to run when object is picked.

7 FETCH OBJECT

7.1 DESCRIPTION

The Fetch Object gets its transform by copying the transform of another object. This makes it easy to get the transform of an object contained in a subnet, and use it as the parent of an object at another level in the object hierarchy.

7.2 PARAMETERS

FETCH OBJECT

Name of the object to use as the source for this object's transform.

FETCH WORLD TRANSFORM

If this option is set, the world transform of the object chosen above will be used as the world transform of this object. If not set, the world transform of this object will be the local transform of the chosen object times the world transform of the parent of this object.

USE / DISPLAY COLOR *dcolorr dcolorg dcolorb*

The display color of the object.

VIEWPORT SELECTING ENABLED

Object is capable of being picked in Viewport.

SELECT SCRIPT

Script to run when the object is picked.

8 FORCE OBJECT

8.1 DESCRIPTION

The Force Object is a place holder for parameters that are used by a number of POPs. It allows you to use direct manipulation in the viewport to control, from the object editor, some of the forces in a POP network.

The Force object is created automatically when using the RBD Operation.

8.2 PARAMETERS – TRANSFORM PAGE

Essentially, the transforms set in the Transform determine the magnitude and direction of the force in the POP objects.

The transform parameters alter the translation, rotation, and scale of the Force object. For a more detailed discussion of these parameters, please refer to the discussion *Translate / Rotation / Scale* p. 256.

8.3 PARAMETERS – FORCE PAGE

TYPE

Force	A simple force along a vector.
Fan	A radially directed fan-like force.
Constraint	Contstrains movement based on the parameters specified below.

FAN ANGLE / DELTA

Determines the directional vector the fan force blows, as well as the spread of the fan force.

CONSTRAINT TYPE

Null	No constraint.
Point	Point constraint
Rotation	Rotational constraint.

The operation of these types, and how they work with the rest of the parameters on this page is discussed below:

null

A Null constraint holds an object in a fixed position in space. There are two parameters required to specify this constraint. The *Nail Point* is the point in world space that the object is constrained to. The Object Point specifies the point on the object (relative to the particle position) that should be constrained to the specified point in space.

So if the Nail Point is specified as (1,1,1), and the Object Point is specified as (0,0,0), forces will be applied to the particle to move it to (1,1,1). Notice that this constraint says nothing about the orientation of the object. If the Nail Point is specified as (1,1,1) and the Object Point is specified as (1,0,0), forces will be applied to the particle so that the particle is one unit distance from (1,1,1), and the object will be oriented so that a vector of (1,0,0) drawn from the particle position would end at (1,1,1). This type of constraint can be visualized as a solid rod connecting a point in space to a point on the object.

point

A Point constraint holds one object in a fixed position relative to another object. This type of constraint requires you to specify a point relative to the position of the first object, and a point relative to the position of the second object (Object A Point and Object B Point). This constraint is similar to the Point to Nail constraint except instead of a nail, the object is fixed to another object. This constraint can be visualized as a rod connecting two objects.

rotation

A Rotation constraint holds an object in a particular orientation. This constraint is specified with a vector in world space (the Rotation Vector) and a vector specified relative to the object (the Object Vector). Torque is applied to the object so that these two vectors remain parallel. This constraint does not in any way affect the position of the particle, only the orientation. Also, the object will still be able to spin freely around the axis specified by the Object Vector.

8.4 PARAMETERS – RENDER PAGE

For a more detailed discussion of each of the parameters associated with this tab, please refer to *Parameters – Render Page* p. 283.

8.5 PARAMETERS – MISCELLANEOUS PAGE

The *Miscellaneous* parameters control variables like the display color of the geometry in the Viewport and the object's selectable status. For a more detailed discussion of the parameters associated with this tab, see *Parameters – Misc Page* p. 261.

9 GEOMETRY OBJECT

9.1 DESCRIPTION

A geometry object is an object that exists in 3D space and is one of the types described in *Geometry Types* p. 223. Geometry objects can be created through a network of SOPs in the SOP Editor, drawn or traced in the Model Editor, or read in from disk with a File SOP. To get to the SOPs contained within an object, select *Edit SOPs* from the tile's pop-up menu.

9.2 PARAMETERS – TRANSFORM PAGE

The *Transform* parameters alter the translation, rotation, and scale of the object. For a complete discussion of these parameters, please refer to the discussion *Translate / Rotation / Scale* p. 256.

9.3 PARAMETERS – SHADING PAGE

SHADER SPACE

The *Shader Space* parameter allows you to select the texture coordinates of another object and apply them to the current object--overriding the existing texture coordinates if any. Selecting the texture coordinates of another object creates a continuous application of a particular shader across the two, or more, objects. For example, in a wallpapered room, the pattern on a light switch housing seldom matches the pattern on the wall. This is analogous to two objects that don't share the same texture coordinates. If both shared the same coordinates, i.e. the light switch was using the wall objects' shader space, the pattern on both the wall and the light switch housing would be the same.

DISPLACE SPACE

Selects the transform space for displacement shading. The pop-up menu associated with this parameter displays a list of objects available for defining displacement shading space. *Displace Space* defines where the shading will occur during the rendering.

SHOP SURFACE

Defines the SHOP to apply as a surface shader.
This shader is used when rendering. The following algorithm is used:

1. If a valid RenderMan SHOP is assigned to the object, it will use this to generate the RenderMan shader.

2. If a valid RenderMan Shader is specified, this will be output to the RIB stream

SHOP PHOTON

Defines the SHOP to be used for photon map generation.

SHOP DISPLACEMENT

Sets a displacement shader for VEX Mantra or RenderMan.

LIGHT MASK

You can select which lights will illuminate a particular geometry object. For example, you may wish to reflection map a logo but not the table it sits on, or you might want to add a colored highlight to the table without affecting the logo. The more lights that shine on an object (especially environment maps) the longer the rendering time, even if the light has little visible effect.

By default, all lights are on for an object. If * appears in the *light mask* text string of a Geometry object's *Shader* parameters, it means all lights illuminate the surface.

When specifying light masks, it is also possible to use object groups. To specify an object group in the lightmask, simply use the at (@) sign before the group name. Groups can be mixed with other patterns. For example:

```
@red_lights,@blue_lights,ambient*
```

would set the light mask for the object to contain all the objects in the groups *red_lights* and *blue_lights* as well as all objects which match the pattern *ambient**.

REFLECTION MASK

Defines the objects which will be picked up in reflections off the current object. Therefore, it is possible to have one object reflect only specified objects rather than the whole scene.

The Reflection Mask parameter allows you to specify object groups. The syntax for groups is the same as described above in the description of *Light Mask*.

Object reflection masks are used to determine ray-tracing subsets in RIB generation.

SMOOTH SHADING

The *Smooth Shading* option enables the interpolation of normals across polygonal surfaces. This creates a smoother surface for your geometry.

BACKFACE REMOVAL

The *Backface Removal* option eliminates the "back" of any geometry in your scene. The backface of a piece of geometry is defined by the geometry's surface normal relative to the position of the camera object. For example, if the camera object were facing a cube dead on, the surfaces of the cube facing away from the camera would not be rendered. If you moved the camera object to the other side of the cube the surface normals facing the camera would be considered the front of the object.

AUTO-GEN REFLECTION MAP

Enabling this will generate a Reflection map for the object. The reflection map will render six images which are then used to create a cubic reflection map around the object specified. This works for both *mantra* and RenderMan.

Once the Reflection Map has been generated, you can disable the button for subsequent renders, and simply specify the generated map in the *Map Name* parameter (below).

map name

This is the name of the reflection map to generate. Since the renderer needs to render six additional maps, this name will be used as a base name for the other images. The other images will have the letters, “xp”, “xm”, “yp”, “ym”, “zp”, and “zm” added to the file name (for the direction that the image represents – i.e. “xp” = X axis, positive direction; and ym = Y axis, minus direction).

map resolution

This is the resolution the images will be rendered at.

NO SHOP TAB

If files are loaded from old versions of Houdini (which used Materials instead of SHOPs), then for backwards compatibility, the Material will be listeted here.

RIB TAB

rib colors

Use these to define what *Cs* and *Os* are in the RIB stream without binding a material to the object. Each object has parameters to specify "RIB Cs" and "RIB Os".

rib shader / rib displace

You can also specify the RIB Shader and RIB Displace here.

MENTAL RAY TAB

You can specify parameters specific to Mental Ray here, including:

SHOP Shadow, SHOP Volume, SHOP Background, SHOP Photon, SHOP Photon Volume, and SHOP Contour.

SOUND TAB

sound material

Reference to a CHOP that defines the sound response of this geometry (see the Spatial Audio CHOP for help).

9.4 PARAMETERS – RENDER PAGE

The Render page allows you to control the parameters for the object during the rendering of the scene. In effect, you are describing to the renderer how the object will behave and how it will appear when rendered.

GEO INSTANCE

Specific to the Geometry object, this parameter creates an instance, or 'copy' of another object's geometry. Instancing differs from copying in that no additional memory is consumed in displaying or rendering the duplicated geometry. The instanced geometry can have transformations applied to it which are independent of the original. Also, the instance can have a separate material applied to it (unless the original has its material defined by a Material SOP (see: *Geometry > Material OP* p. 634)). It also accepts separate shading parameters.

Geometry Instancing is supported by both *mantra* and RenderMan. This means that rendering consumes much less memory than it would if the geometry's data were actually duplicated. For example, by instancing one tree many times, a forest can be rendered while only using the geometric memory needed for one tree.

POINT INSTANCING

If there is a particle system in this object, the geometry object specified here will be "instanced" to each particle. Rather than making copies of the geometry at each particle, this provides a light-weight (less memory usage) method of adding complexity to the scene.

This parameter allows you the most efficient way to animate many copies of geometry with a particle system. Make one object the particle system, and another the geometry to instance. Then select the geometry object in the particle system object's *Render* page under *Particle Geo*. Your geometry will be instanced to each of the particles in the system without consuming additional memory for each of the copies.

If an up vector exists for the particles, it will be used to orient the instanced geometry. You create and control up vectors for particles by using a Point SOP.

Production Tip: Create a one-particle particle system, then copy it to template geometry. This will allow you to instance geometry without having to create many objects.

point geometry

Specify the geometry to be instanced here.

GEOMETRY

Geometry as is Renders the geometry as you see it.

Polygons as Subdivision Surfaces
Use geometry from Render SOP, but render polys as sub-division surfaces. Valid only for Mantra/RenderMan.

Render only points Renders only points and not surfaces. Useful for quick test renders. Use geometry from Render SOP, but render only point primitives at each point. A point primitive will be rendered at each point of the SOP geometry. The "width" attribute can be used to specify the size of each point. If no "width" attribute exists, the default size is 0.02.

Render from File This causes *mantra* to load the geometry from a disk file instead of sending the geometry data in-line in the IFD. This can dramatically reduce IFD generation times. See below*.

Bounded file (explicit bounds)
In this case, along with a geometry file to load, a bounding box for the geometry (in SOP space) is given. This allows mantra to defer the loading of geometry until the object is actually rendered. This can provide large memory savings because mantra won't load any geometry which isn't rendered. The bounding box specified MUST enclose the geometry produced. It can be larger than the actual geometry, but for efficiency, it should be as close to the actual size of the geometry as possible. See below*.

Bounded file (BBox from Render SOP)
Use geometry from file, but let mantra/RenderMan use deferred loading of geometry files. Instead of specifying the bounding box by hand, let Houdini compute it using the bounding box of the render SOP – see below*.Subdivision from file - Use geometry from file specified, render polygons as subdivision surfaces. This is only valid for mantra.

Subdivision from file Use geometry from file, rendering polygons as subdivision surfaces.

Subdivision from file (explicit bounds)
Use geometry from file, rendering polygons as subdivision surfaces. Specify bounding box for deferred load explicitly. This is only valid for mantra.

Subdivision from file (render SOP)
: Use geometry from file, rendering polygons as subdivision surfaces. Use render SOP to compute bounding box. This is only valid for mantra.

Points only from file
: Use geometry from file. Render point primitives only. This is only valid for mantra.

Points only from file (explicit bounds)
: Use geometry from file, rendering only points. Specify bounding box explicitly for deferred load. Only valid for mantra.

Points only from file (render SOP)
: Use geometry from file, rendering only points. Use the bounding box of the render SOP. Only valid for mantra.

* rendering from a file

When specifying disk files for *mantra* to load geometry from, it is also possible to specify an external program to generate geometry. This is done by prefixing the command (and arguments) with a single pipe symbol (|). For example:

```
|gconvert somefile stdout.bgeo
```

Mantra will read the geometry from the output of the program on a pipe. Again, the bounding box specified must enclose the geometry generated by the pipe command.

MANTRA PROCEDURE

Procedural geometry for mantra. Mantra will call this procedure to generate render geometry.

RIB PROCEDURE

Specify an RiProcedural (instead of SOP geometry).

SHOP GEO SHADER

This parameter is used to specify a Mental Ray geometry SHOP for the object. Currently (v 6.0), only Mental Ray supports geometry shaders.

MOTION BLUR

You can apply several different kinds motion blur effects to an object for rendering:

none

Disables motion blur on the object.

inherit behaviour

This option specifies that the object will inherit the motion blur specifications from its parent object.

transformation blur only

Transformation motion blur uses only the transformation on the object (including transformations of the parent object) to compute motion blur. In this case, the geometry is the same at the beginning of the frame and at the end of the frame.

deformation blur

This option blurs an object's deformations as specified in the Surface Operations Editor. *Deformation Blur* includes *Transformation Blur*.

Deformation motion blur computes the geometry at the beginning of the frame, and at the end of the frame to generate two different geometries to blur between. This has bad side-effects with simulations which cannot move backward in time. For example, if a Particle OP has geometry copied to it, the simulation will be cooked at the beginning and end of the frame. This means that if particles are birthed mid-frame, there cannot be motion blur applied because the topologies are different. Deformations within an object are usually achieved by the SOPs within the object.

Note: In order for motion blur to render, you must disable Variance Anti-aliasing in the *mantra* Output (i.e. eliminate the *-v* option from the *Render Command* string.

velocity attribute blur

Velocity attribute motion blur is similar to deformation motion blur, except that the velocity attribute (on points) is used to compute the deformation on the geometry. Using this type of motion blur ensures that the particle simulations are *not* cooked ahead in time, meaning that motion blur should always work correctly.

DISPLAY

If the *Display* option is enabled, the value in the edit field overrides the object's display flag (the blue rectangle on the object tile). Lights and Cameras will never appear in the rendered image.

MATTE

Selecting the *Matte* option for an object will exclude it from the rendering process and create a cut-out area where the object was. Objects behind the matted selection will be rendered.

PHANTOM

Phantom also excludes the object from the rendering process but specifies that the object's effect on a scene (the shadows it casts or its reflection) will be rendered.

REFLECT / REFRACT BOUNCE

The *Reflect and Refract Bounce* options designate the number of times a ray is reflected or refracted during a render. For example, if the camera in a scene has a mirror in front of it and a mirror behind it, setting the *Reflect Bounce* option limits the amount a ray bounces off the mirrors. The default value for this parameter is 10.

RAY CLIP

Defines the maximum color level for ray propagation from the surface of the object. This is similar to ray bounce in that it can be used to terminate ray-tracing early.

For example, by setting the Ray Clip to 0.1, no reflection rays will be sent if the net contribution to the resulting color of the pixel will be less than 10% (0.1). This is often a much better algorithm than ray-bounce for limiting ray-tracing.

DISPLACE BOUND

When you have a displacement map with *vmantra*, mantra doesn't know what the surface will look like until it shades the displacement (i.e. applies the displacement shader to the geometry). Once this happens, then mantra can resolve where the surface goes. Mantra tries to minimize the amount of work it does by sorting geometry into the tiles where the geometry appears. However, with displacement shaders, it's possible that the geometry could be moved somewhere before it's time for mantra to render the geometry. This means you have to give mantra an idea of how much the geometry can move before it renders, so that mantra can put it in the tiles which it might be moved to. The problem is that if your geometry can be moved a lot (relative to the screen position), mantra will end up thinking that there's a lot of geometry in every tile, meaning that it has to do more work, meaning that the rendering time increases substantially.

RenderMan displacement bounds define the maximum range that a displacement map can move geometry. In cases where cracks appear in the displacement maps, or if there are bad rendering problems, you should increase this value. For more information, please consult Pixar's documentation.

SHADING QUALITY

The *Shading Quality* parameter activates the three parameter fields below it and sets the quality with which the lighting model (i.e. material) is applied to the object. This means the material applied to the object can have its sharpness increased or decreased depending on how close your camera is to that object (enter an expression that changes the *Shading Quality* based on distance from the camera).

Decreasing this value speeds up the rendering process, but also decreases the sharpness on which that material is applied. Increasing the *Shading Quality* improves the material's texture map, bumps, and dent quality, but also increases the rendering time.

The most important thing to remember is that Shading Quality applies to the material in general. If there are textures or bumps, dents or erodes in the material, the *Shading Quality* value has a tremendous impact on the success of the render.

You therefore want to adjust the *Shading Quality* for a couple reasons:

- Texture maps with text/font images or texture maps used as decals need a higher level of *Shading Quality* to improve the quality of the material and avoid aliasing in the textures.

- Texture maps with dents, bumps, or erodes with a high frequency need a higher *Shading Quality* setting in order to avoid flickering.

If the *Shading Quality* is not set high enough you will get noise or aliasing artifacts in your texture map, no matter what your Super Sampling is set to. It can be thought of as *Super Sampling* for your materials (lighting models).

range of values for shading quality

0.01	Extremely low
5	Extremely high (increases render time about 25 times)
1 to 2	Nominal.

Note: The *Shading Quality* parameter is available both per-object (object > *Render* page > *Shading Quality*), and Globally (Output Editor > mantra3 > *Render Command* ✣ > *Micro Polygon* sub-page > *Shading Quality*). These two values are multiplied to determine the final Shading Quality used in the render.

LEVEL OF DETAIL

Level of Detail affects the quality of rendered (especially NURBS) geometry. Lower values produce a coarser looking surface while higher values produce a more highly refined surface. Level of detail changes are more noticeable in motion than for a static frame.

note – on level of detail vs shading quality

mantra has two modes of rendering (*mantra3* has equivalent modes). Micropolygon uses *Shading Quality* to adjust the size of micropolygons. Ray-Tracing (and scan-line rendering in *mantra*) use the LOD as a similar control (i.e. they will sub-divide primitives based on the LOD). Thus, you can think of *LOD* as the ray-tracing equivalent of *Shading Quality*. Basically, you can have a lower LOD on a surface which means it will be coarser in ray-tracing, while keeping the primary rays at a high quality.

CAUSTICS MODE

Used soley by Mental Ray to determine whether the surface will cast or recieve caustic photons.

GLOBAL ILLUMINATION MODE

Used solely by Mental Ray to determine whether the surface will cast or recieve global illumination photons.

PRE / POST INCLUDE (ALL TYPES)

The *Pre-Include* is included before an object/camera is defined. The *Post-Include* is defined afterwards (i.e. after the transform and geometry definitions, and just before the AttributeEnd). For each renderer this is slightly different (and some renderers don't even support this).

example

If (instead of specifying it with the *White Point* parameters in the Mantra Output OP), you wanted to set the white-point during the render, you could put:

```
ray_whitepoint 0.5
```

in the *Camera Pre-Include*, which would have the effect of mapping full illumination to 0.5. This allows colors brighter than 1 to be rendered into the image.

mantra

Camera Pre-Include	After the "world" information has been defined. This allows you to alter things like the projection, bucket size, output picture etc. Additional geometry can also be defined here.
Camera Post-Include	Just before the rendering begins. This allows you to include additional objects or materials before the render begins.
Object Pre-Include	Before the object is even defined
Object Post-Include	After all the object definition commands. It is possible to augment the transform of an object or set some parameters.
Light Pre-Include	Before the light is defined (after the comment for the light).
Light Post-Include	After the light has been defined. It is possible to change the shader or transform of the light here.
Atmosphere Pre-Include	Before the atmosphere object is defined
Atmosphere Post-Include	After the atmosphere object is defined

renderman

Camera Pre-Include	After the *RiFrameBegin()* call, and after all the parameters for the camera have been set up, but before the camera transform and WorldBegin.
Camera Post-Include	Just before the WorldEnd. This is before the FrameEnd of course.
Object Pre-Include	After the AttributeBegin and the Attribute "identifier", but before any shaders, colors or geometry.

Object Post-Include	Between the Shader output and the Geometry definitions. This is before the transform for the geometry.
Light Pre-Include	Before the light has been defined.
Light Post-Include	After the LightSource statement (i.e. after the transform and LightSource have been defined).

9.5 PARAMETERS – PHYSICAL PAGE

ACTIVE RIGID BODY

Specifies whether the object should be an active or passive rigid body.

MASS

Mass of the object. A mass of zero is equivalent to an infinite mass.

BOUNCE

Bounce coefficient of the object. When two objects collide, the product of the bounce coefficients of the objects determines the energy lost to the collision (and thus the velocities of the object after the collision).

DYNAMIC FRICTION

Coefficient of dynamic friction. When two objects are in contact and sliding across each other, the product of their dynamic friction values determines the force of friction acting against the sliding motion.

STATIC FRICTION

Coefficient of static friction. When two objects are in contact and relatively at rest, the product of their static friction values determines the amount of force required to induce a relative (sliding) motion of the objects.

CENTRE OF MASS

Specifies the location of the objects center of mass. This value is specified as a displacement in object space from the origin (0,0,0) of the object. The rigid body particle representing an object is always placed at the origin of the object, which may not be the real center of the geometry. In this case you would specify a center of mass so that the particle behaves as if it's center of mass is at the center of the geometry.

LINEAR VELOCITY

Velocity of the object on the first frame of the simulation.

ANGULAR VELOCITY AXIS

Angular velocity of the object on the first frame of the simulation. This parameter specifies the axis around which the object is rotating.

ANGULAR VELOCITY RATE

Angular velocity of the object on the first frame of the simulation. This parameter specifies the rate at which the object is rotating around its axis of rotation.

POP GEOMETRY

Specifies the SOP to use for collision detection by the rigid body solver. This parameter is also used to specify the guide geometry that should be used to represent a particle with this object as its instance attribute.

COLLISION GEO LOD

If the collision geometry specified for this object is not made of triangles, it must be converted to triangles internally by the rigid body solver. This specifies the LOD that should be used to convert NURBS and Bezier surface.

INERTIAL TENSOR

This parameter specifies the method that should be used to calculate the inertial tensor of the object. The inertial tensor is a matrix that describes how the object reacts to torque. It is a function of the distribution of mass through the object, and its shape. The inertial tensor can be calculated either by counting each point in the geometry as having identical mass, or by picking random sample points within the object, and assigning equal mass to each of them. The first method is faster and allows greater control (you can specify exactly where the mass of the object is located). The second method is simpler to use, and provides realistic looking results without mush effort.

samples

If using the random sample point method to calculate the inertial tensor of the object, this parameter specifies the number of sample points that should be used.

GEOMETRY CHANGES WITH TIME

If this parameter is set, the collision geometry and the inertial tensor are re-calculated at each frame. Otherwise these values are only calculated for the frame in which the particle is created. Turning this parameter on will greatly slow down the simulation, but is the only way that accurate collision detection can be performed on deforming geometry.

9.6 PARAMETERS – MISCELLANEOUS PAGE

The *Miscellaneous* parameters control variables like the display color of the geometry in the Viewport and the object's selectable status. For a more detailed discussion of the parameters associated with this tab, see *Parameters – Misc Page* p. 261.

10 HANDLE OBJECT

10.1 DESCRIPTION

The Handle Object is a new IK tool designed for manipulating bones. Whereas the previous IK tools only allowed for a single end-affector per bone chain, this new method allows for several end-affectors per bone. Furthermore, the bones need not be a chain. Any setup, including branches, are handled.

One typical example is the use of motion capture data. In higher-end systems, you typically have a cloud of marker positions and a skeleton to be driven by it.

This Handle Object works in tandem with the Handle CHOP (see for example). The following setup is commonly used: Given a hierarchy of bones, attach one or more Handle objects to specific locations on each bone object. Assign each handle a target in space to follow. Create a Handle CHOP which collects this information and calculates the rotation channels for the bones. Export these values back to the bones.

10.2 PARAMETERS – TRANSFORM PAGE

TARGET

Where the Handle will pull the bone towards. (like an end-affector).

OFFSET

Displacement tx,ty,tz relative to the origin of the bone where the handle is anchored.

WEIGHT

When multiple handles are attached to a bone, higher weighted handles draw closer to their target.

TWIST ONLY

When enabled the handle only contributes the the z rotation of the bone, when disabled it points the bone (rx, ry, and rz).

FALLOFF

This value affects how much parent bones are affected when the child bone is reaching a target. It affects the flexibility of the chain.

ROTATION LIMITS

These affects how much the bone is allowed to rotate with respect to its parent.

11 LIGHT OBJECT

11.1 DESCRIPTION

Light Objects are those objects which cast light onto other objects in a scene. With the light parameters you can control the colour, shadows, atmosphere and render quality of objects lit by the light. Lights can also be viewed through and used as cameras (Viewport > *Camera* menu).

11.2 PARAMETERS – TRANSFORM PAGE

The *Transform* parameters alter the translation, rotation, and scale of the object. For a detailed discussion of these parameters, please refer to the discussion *Translate / Rotation / Scale* p. 256.

11.3 PARAMETERS – SHADING PAGE

SHOP LIGHT

Specify the Illumination shader for rendering.

SHOP SHADOW

The Shadow / occlusion shader for rendering.

SHOP EMITTER

Used by Mental Ray (only) to emit photons.

SHADOW MASK

Determines which objects cast shadows from the light source. For example, if there are two objects (geo1 and geo2) sitting above a ground plane, it is possible to set the *Shadow Mask* to "geo1" and have only geo1 cast shadows (even though both objects are lit). Use * to specify all lights.

When specifying shadow masks, it is also possible to use object groups. To specify an object group in the shadow mask field, simply use the at (@) sign before the group name. Groups can be mixed with other patterns.

DEPTH MAP

The style of depth map to generate from this light source. Mid-Point Z-depth shadows help to eliminate self-shadowing artifacts, but do not accurately represent the distance to the occluding object and so should not be used for atmospheric lights. Currently only RIB generation supports deep shadow map generation.

Z-DEPTH MAP

This edit field supplies the name of the Z-Depth map to auto-generate (if enabled), or the filename of an already existing Z-Depth map. Select a file name for the depth map by typing the location of the file into the edit field, or by selecting a map using the directory dialog.

AREA SHAPE

Currently used only by Mantra and Mental Ray to represent area light.

size / samples

When Area shape is enabled, this specifies the size and samples to use.

CONTRIBUTES TO DIFFUSE / SPECULAR

These two parameters determine whether the light will contribute to the diffuse and specular illumination when shading objects. If either of these is disabled, the light will not contribute to the diffuse/specular illumination.

This can be used to eliminate highlights on a surface being lit by multiple light sources. For example, it is possible to have a “fill” light which just fills in diffuse light without adding an additional specular highlight on the surface. Alternately, it can add a highlight without adding anything to the diffuse illumination on a surface.

These parameters affect only the Phong, Blinn, and Cook lighting models. The diffuse option can be used to affect Lambert shaders also, but it is better to turn the light value to 0 in this case to reduce shading computation.

PHOTONS CONTRIBUTE TO DIRECT ILLUMINATION

When emitting photons from a light source, mantra will typically force photons which hit diffuse surfaces to bounce at least one time. This allows photon maps to be used for secondary illumination without having to pull shader tricks. Enabling this parameter causes photons to be stored in maps even on the first intersection with other surfaces.

USE FAR CLIPPING PLANES AS ACTIVE RADIUS

When the scene contains a large number of light sources, shading can get bogged down when commuting illumination. This option allows mantra to ‘cull’ lights which are farther than the far clipping plane away from the surface being shaded. Caution should be used with this parameter, since without careful use of attenuation, shading discontinuities may be generated.

11.4 PARAMETERS – SHADING PAGE > NO SHOP TAB

LIGHT COLOUR

You can modify the color of a light three ways: using the Color List dialog (click on the ✣ button), using the Hue, Saturation, and Value (HSV) controls, or with the Red, Green, and Blue (RGB) controls. To switch between the HSV and RGB controls, click on the appropriate colour box and the color ranges below will change accordingly.

hue / saturation / value

This mode of color definition allows you to specify the Hue, the Saturation and the Value of a given color. *Hue* describes a color's place on the spectrum. *Saturation* defines the amount of white mixed with the color, while *Value* describes the lightness or darkness of the color. You can edit the values by dragging the "thumb" of the slider bar, or by manually entering the values in the edit fields.

red / green / blue

This mode of color definition allows you to specify the Red, Green, and Blue values that are mixed together to form the color you want. Full (1.0) values for Red, Green, and Blue yield white; while zero (0.0) values yield black. You can edit the amounts by dragging the slider, or by manually entering the values in the edit fields.

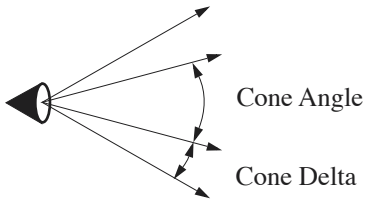
DIMMER

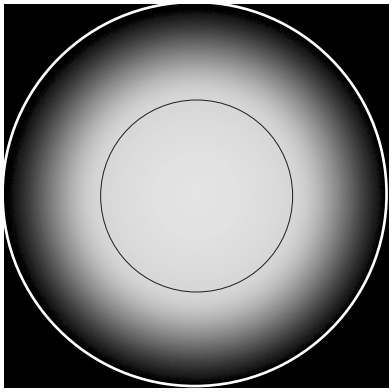
This parameter changes the intensity of the light without affecting its hue.

LIGHT TYPE

Choose between *Point Light* and *Cone Light* types.

CONE ANGLE





Enabling this option sets the type of light to conical. Use the edit field or angle control icon to designate the size of the cone angle in degrees. The *Cone Angle* specifies the angle within which the light remains at full intensity. Decreasing the cone angle to between ten and forty degrees focuses the beam to spotlight proportions.

CONE DELTA

This value, in degrees, represents the angle outside the cone angle through which the light intensity drops from its maximum to zero. Beyond this area, no more light is cast.

CONE ROLLOFF

This parameter (a value between one and ten) defines how gently or suddenly the amount of light decreases between full intensity and zero intensity within the Cone Delta area.

SHADOW SHADER

This determines the type of Shader the light will use to cast Shadows. Entering "noShader" in this field will turn off shadows for the light. Entering one of:

- fastShadow
- transShadow
- filterShadow
- zShadow

will turn on Shadows for the light, using the respective shader to create the shadows. Clicking the ✣ button displays a dialog script where you can set these. For a discussion of the parameters available in the dialog script, see *Light Shader ✣ Dialog* p. 360.

ATTENUATION

Sets the distance at which light drops to half intensity compared to the intensity at the light source. Intensity drops at the unrealistic rate of *1/(1+r)*. By default, the light intensity does not fade as it travels.

SHADOW SOFTNESS

By moving the slider control, you can select the degree of softness for shadows in the scene. The higher the value, the softer the shadows cast. Shadow Softness only applies if you have *Enable Blur* turned on in the light's *Shader* ✣ parameter (*Shading* page > *Enable Blur* parameter in *Light Shader ✣ Dialog* p. 360).

REFLECTION MAP SCALE

An environment light source can illuminate the scene from all directions. The reflection vector (from the eye) is traced to the environment map. This is often called Reflection Mapping.

This parameter lets you determine the horizontal and vertical scale of the reflection map. The reflection map is set in the *Shading* page parameters, see below.

REFLECTION MAP

Reflection maps compute the way a surface reflects its environment. Already having a reflection map reduces rendering time by eliminating the need to re-raytrace a surface's reflection of its environment. You can enter the name of the map in the edit field or select a file using the directory dialog–activated by clicking on the ✣ to the right of the edit field.

Reflection maps are scaled by the specular component of the material.

Tip: For Reflection Maps, the light should be pointing at the object, and the light source should be pointing down the Z-axis.

DIFFUSE MAP

The *Diffuse Map* parameter is similar to the *Reflection Map* parameter, but affects only an object's diffuse color component.

PROJECTOR MAP

The *Projector Map* parameter allows you to specify an image to be projected onto a scene or geometry surface. The principle behind the *Projector Map* is similar to a slide projector. The image is "placed" in front of the light source and shines onto the scene or geometry.

Tip: You can simulate a faster version of a cone light by providing a simple Round projection map that fades off at the edges, say by creating a Radial-Concentric ramp in the COP Editor, or by using the image *$HH/pic/circle.pic* .

ATMOSPHERE

The *Atmosphere* parameter for light sources can be used to render volumetric light sources. Currently, there is only one such shader, *lbeam*. This creates a beam of light which covers the field of view of the light source. Clicking the ✣ button beside the Atmosphere parameter displays a Dialog script where you can set numerous parameters relating to Atmosphere. See *Atmosphere ✣ Dialog* p. 363 for details on these.

11.5 PARAMETERS – SHADING PAGE > RIB TAB

RIB SHADER

The dialog box associated with this parameter allows you to select from a range of RenderMan shader types. The parameters below update according to your selection.

If you leave the RenderMan shader for the light blank, Houdini should provide the correct shader for your light (i.e. if you look at the RIB, it should contain an entry something like "attenuation" within it).

If you've chosen "shadowpoint" or one of the other RenderMan light shaders, be sure to specify the map name in the shader.

11.6 PARAMETERS – SHADING PAGE > MENTAL RAY TAB

ENERGY

This is the energy used for photon map generation (for either caustic or global illumination).

EXPONENT

This parameter determines the exponential falloff for photons emittted from the light source for caustic or global illumination photon mapping.

CAUSTIC PHOTONS

The number of photons to store/emit for caustics.

GLOBAL ILLUM PHOTONS

The number of photons to store/emit for global illumination.

11.7 PARAMETERS – VIEWING PAGE

For a discussion of the *Viewing* parameters, please refer to *Parameters – View Page* p. 271.

11.8 PARAMETERS – RENDER PAGE

The *Render* parameters control the object's behaviour and appearance when rendered. For detailed discussion of the parameters associated with this tabbed page, please see *Parameters – Render Page* p. 283.

11.9 PARAMETERS – MISCELLANEOUS PAGE

The *Miscellaneous* parameters control the Light object's selectable status and allows you to select a script that will run when you select the object. For a detailed discussion of the parameters associated with this tabbed page, please see *Parameters – Misc Page* p. 261.

12 MICROPHONE OBJECT

12.1 DESCRIPTION

The Microphone object specifies a listening point for the SpatialAudio CHOP. Multiple microphones can be used by one SpatialAudio CHOP to create stereo or surround sound, with special effects like the doppler effect, volume loss over distance, obstacle interference, atmospheric filtering and positional audio.

To set up a Spatial Audio scene, one or more Sound objects should be used to emit sound. At least one microphone is needed to capture the sound. A SpatialAudio CHOP is needed to render the sound. If any obstacles or filters are used, at least one Acoustic CHOP is needed to design the spectrum filter.

Moving Sound and Microphone objects around will produce variations in pitch and volume, especially if either object is directional. Setting up a directional microphone or sound object is much like setting up a directional light.

Note: See *The 3D Spatial Audio System* p. 367 for a synopsis of Houdini's Spatial Audio System.

12.2 PARAMETERS

MICROPHONE ACTIVE */micactive*

Turns the microphone on (if greater than zero) or off.

12.3 PARAMETERS – MICROPHONE PAGE

SENSITIVITY */sensitive*

The volume gain of the microphone.

DIRECTIONAL

When disabled, the microphone is non-directional and accepts sounds from all directional equally.

When enabled, the microphone is directional and pointed down the negative Z axis. The *Recording Cone* and *Outer Cone* parameters determine the field of recording.

Any sounds within the recording cone have a gain defined by the *Sensitivity* (above). Any sounds outside the outer cone have a gain defined by the *Outer Sensitivity* parameter (below). Any sound between the recording cone and the outer cone have a gain that is interpolated between these two sensitivities.

RECORDING CONE *reccone*

Defines the angle of the recording cone, measured from the Z axis to the edge of the cone. Setting this to 180 degrees will make a non-directional microphone.

OUTER CONE *outcone*

Defines the dropoff region from the recording cone to the outer cone. If the recording cone is set to 60 degrees and the outer cone is set to 20 degrees, any sounds coming from more than 80 degrees off the Z axis will fall outside the outer cone.

DROPOFF

Sets the interpolation type for the dropoff from the recording cone sensitivity to the outer sensitivity.

DROPOFF RATE *droprate*

Increases or decreases the rate of dropoff.

OUTER SENSITIVITY *outsensitive*

The gain applied to any sound coming from outside of the outer cone area.

FILTER

Specifies an optional CHOP that defines the frequency response of the microphone. The filter should be a channel named 'absorb', created with an Acoustic CHOP.

12.4 PARAMETERS – MISC PAGE

For a complete description, see *Parameters – Misc Page* p. 259.

DISPLAY COLOR

The colour to use when displaying the object in the Viewport.

VIEWPORT SELECTING ENABLED

Allows the geometry to be selected in the Viewport

SELECT SCRIPT

The script to run when object is picked.

13 NULL OBJECT

13.1 DESCRIPTION

The Null Object serves as a place-holder in a scene, and does not render. It can be used to designate a place in the space of your scene, or as a "look at" object to help coordinate a camera's movement and field of view.

The Null Object is also used as an End Affector for constructing chains of bones. Its parameters are similar to those of the Geometry Object's *Transform* and *Miscellaneous* pages.

13.2 PARAMETERS – TRANSFORM PAGE

The transform parameters control the object's movements and orientation in space. They also define its relation to other objects through the *Look at* and *Path* parameters. For more information see *Parameters – Transform Page* p. 256.

13.3 PARAMETERS – MISCELLANEOUS PAGE

The Miscellaneous page allows you define an object's display color and selectable status. For more information see *Parameters – Misc Page* p. 261.

14 SOUND OBJECT

14.1 INTRODUCTION

The Sound object defines a sound emission point for the Spatial Audio CHOP. Multiple sound sources can be mixed by one Spatial Audio CHOP to create stereo or surround sound, with special effects like the doppler effect, volume loss over distance, obstacle interference, atmospheric filtering and positional audio.

To setup a Spatial Audio scene, one or more Sound objects should be used to emit sound. At least one microphone is needed to capture the sound. A SpatialAudio CHOP is needed to render the sound. If any obstacles or filters are used, at least one Acoustic CHOP is needed to design the spectrum filter.

Moving Sound and Microphone objects around will produce variations in pitch and volume, especially if either object is directional. Setting up a directional microphone or sound object is much like setting up a directional light.

Note: See *The 3D Spatial Audio System* p. 367 for a synopsis of Houdini's Spatial Audio System.

14.2 PARAMETERS – SOUND PAGE

SOUND ACTIVE *soundactive*

Turns the sound source on (if greater than zero) or off.

VOLUME *volume*

The volume of the sound source.

DIRECTIONAL

If off, the sound source is non-directional and emits sound equally in all directions.

If enabled, the sound source is directional and pointed down the negative Z axis. The emission cone and outer cone parameters determine the field of emission. Any sound emitted from within the emission cone has a gain defined by the *Volume* (above).

Any sound emitted outside the outer cone has a gain defined by the *Outer Volume* parameter (below). Any sound emitted between the recording cone and the outer cone have a gain that is interpolated between these two volumes.

EMISSION CONE *emitcone*

Defines the angle of the emission cone, measured from the Z axis to the edge of the cone. Setting this to 180 degrees makes a non-directional sound source.

OUTER CONE */outcone*

Defines the dropoff region from the emission cone to the outer cone. If the emission cone is set to 60 degrees and the outer cone is set to 20 degrees, any sounds emitted from more than 80 degrees off the Z axis will fall outside the outer cone.

DROPOFF

This menu sets the interpolation type for the dropoff from the *Emission Cone* volume to the *Outer Volume*. Select from: *Linear*, *Ease-in*, *Ease-out*, and *Ease-in-out*. The default in *Linear* – which processes slightly faster, the *Ease* selections will sound more natural.

DROPOFF RATE */droprate*

Increases or decreases the rate of dropoff.

OUTER VOLUME */outvolume*

The gain that is applied to any sounds coming from outside the *Outer Cone*.

SOURCE

Specifies the CHOP that contains the audio that this Sound object is emitted. Only one audio track can be emitted; if there is more than one audio track in the CHOP, the first one is used.

14.3 PARAMETERS – MISC PAGE

For a complete description, see *Parameters – Misc Page* p. 259.

DISPLAY COLOR

The colour to use when displaying the object in the Viewport.

VIEWPORT SELECTING ENABLED

Allows the geometry to be selected in the Viewport

SELECT SCRIPT

The script to run when object is picked.

15 SUBNETWORK OBJECT

15.1 DESCRIPTION

This Object allows for the simplification of complex networks by collapsing several Objects into one. Four inputs are provided to connect Objects in the sub-network with Objects in the parent network.

Sub-network Objects can also be nested within each other.

Objects Subnets contain a couple differences from other subnets: i) instead of using a flag on a contained node to determine the output of the subnet, the subnet has a parameter to specify which contained node to use as the output for the subnet.

ii) the display flag (and the display parameter) of the object subnet acts like a mask for all its children (i.e. if the subnet display is turned off, all children are also not displayed. If the subnet display is on, only those children with their display on are displayed).

15.2 PARAMETERS

INPUT LABEL

These four string parameters allow labels to be specified for the four inputs to the sub-network. These labels appear when the middle mouse button is depressed on the input connections and is useful for describing the use of each input to the sub-network.

DISPLAY *(display*

Specifies whether object is visible. Objects contained in the sub-network are only visible if the containing object is visible.

OUTPUT TRANSFORM

This parameter specifies which object contained in the sub-network should be used as the output transform of the SubNet object.

16 SWITCHER OBJECT

16.1 DESCRIPTION

The Switcher object allows you to switch Camera objects. You do this by wiring multiple Camera objects into its input, and selecting the input camera via the Switch Camera parameter.

This object acts like a regular Houdini camera, except that it gets all of its parameters (including transforms) from the input camera. The switch camera has its own rotoscoping options.

16.2 PARAMETERS – VIEWING PAGE

SWITCH CAMERA

The value here will choose one of the input camera objects in ordinally. Animating this parameter allows you to switch between cameras in the course of an animation.

16.3 PARAMETERS – ROTOSCOPE PAGE

COP NETWORK / COP NAME

Select a COP which to rotoscope.

COP FRAME

Specify the frame to use. Typically set to $F – the current frame.

APPLY ZOOM TO BACKGROUND

Matches camera zooms, and applies them to the rotoscoped image.

16.4 PARAMETERS – RENDER PAGE

The *Render* parameters control the object's behaviour and appearance when rendered. For detailed discussion of the parameters associated with this tabbed page, please see *Parameters – Render Page* p. 283.

Note: The motion blur setting is not inherited from the input camera.

16.5 PARAMETERS – MISCELLANEOUS PAGE

The Miscellaneous page allows you define an object's display color and selectable status. For more information see *Parameters – Misc Page* p. 261.

3 Object OPerations

1 AMBIENT OPERATION

1.1 DESCRIPTION

The Ambient Light operation is used to add ambient illumination to your scene. An Ambient Light lights all objects in a scene evenly. It is generally good to have one Ambient Light in the scene.

Although Ambient Light objects do not have transform parameters, it is still possible to select a parent object for the new light. Once the parent object (if any) is selected, right-click ( ) to complete the operation.

This operation has no handles associated with it, but the color of the light can be specified in the operation controls bar immediately above the viewport.

You can change the Ambient Light's parent object any time by hitting the Reselect Geometry button or typing . If *Secure Selection* is turned off, selecting a new object will immediately complete the selection and connect the new parent. Otherwise, you right-click ( ) to signify that the object selection is complete.

If Secure Selection is turned off, you can select any other object in the viewport. Doing so will switch to the operation appropriate for that object type, and the current operation will be exited immediately.

1.2 SEE ALSO

- *Light Operation* p. 330

2 BLEND OPERATION

2.1 DESCRIPTION

The Blend operation is used to create a new object with a transform that is a blend of transforms from up to four input objects.

You select the objects from which you wish to create the blended transform. Once the objects are selected, right-click (🖱) to complete the operation. You are presented with a transform handle attached to the new Blend object. This handle lets you modify the transform that is applied to the Blend object after the blending of the input transforms.

The operation controls bar above the viewport contains the relative weights to assign to each of the input transforms. By adjusting these parameters it is possible to modify (and animate) the degree to which each input transform affects the final blended transform.

You can specify a different set of parent objects by hitting the Reselect Geometry button or typing ⏎. If Secure Selection is turned off, selecting one or more new objects will immediately complete the selection and connect the new parents. Otherwise, you right-click (🖱) to signify that the object selection is complete.

If Secure Selection is turned off, you can select any other object in the viewport. Doing so will switch to the operation appropriate for that object type. The current operation will be exited immediately.

2.2 SEE ALSO

- *Fetch Operation* p. 324

3 BONE OPERATION

3.1 DESCRIPTION

The Bone Operation is used to create a new Bone object that can be used as part of a bone chain to do forward or inverse kinematics.

You select the object that will be used as the parent for the new Bone object, and right-click (🖱) to complete the operation. The new Bone object appears with handles that can adjust the length of the bone, or its rotation angles. The length and rotation parameters are also available in the operation controls bar above the viewport.

When creating bone chains, it is almost always best to use the Bones operation, which can be used to quickly create entire bone chains and set up kinematic solvers for the bones.

You can change the Bone object's parent object any time by hitting the Reselect Geometry button or typing ⌨. If Secure Selection is turned off, selecting a new object will immediately complete the selection and connect the new parent. Otherwise, you right-click (🖱) to signify that the object selection is complete.

If Secure Selection is turned off, you can select any other object in the viewport. Doing so will switch to the operation appropriate for that object type, and the current operation will be exited immediately.

See the Tutorial Guide > Understanding Bones chapter for more info.

3.2 SEE ALSO

- *Bones Operation* p. 310

4 BONES OPERATION

4.1 MOUSE CLICKS

For a complete list of the mouse controls at all times, please refer to the status line at the top of the Houdini window. The common tasks are described below.

LMB	Start drawing bones.
Shift LMB	Start drawing from object.
Ctrl LMB	Split bone.
Ctrl MMB	Select parent.
Del	Backup one step, and remove the last entered node.
RMB	Finish drawing
Ctrl RMB	Pops up the Operation menu.
Alt LMB	Constrains to direction perpendicular to cPlane or viewplane.

BONE CREATION – SUMMARY

- To begin drawing bone chains, click with LMB where you want the chain to start – you'll see a bone attached to the cursor.
- Click with LMB again to complete a bone and start the next bone.
- Delete the last created bone by typing the Del key.
- To select a parent to attach bones to, select it with Ctrl MMB.
- Click with RMB to commit the entry of your bone chain – a hierarchy of bone objects (and end effectors if applicable) are created for you in the Layout Area.

4.2 DESCRIPTION

See the Tutorial Guide > Understanding Bones chapter for more info.

The Bones operation lets you interactively draw and manipulate bone chains before animating, and it will create an IK CHOP and appropriate bone objects.

Some special features of the Bones Operation include:

- The creation of capture regions, bone objects, and end effectors automatically
- Easy parenting of a new bone chain to other geometry
- Control over the naming of new geometry
- The replacement of standard bone geometry with user geometry
- Explicit numeric specification of bone positions
- Automatic colouring of bone chains.
- Easy positioning of the joints of bones.

DRAWING BONE CHAINS

To draw a new bone chain, left-click ( ) repeatedly in the viewer to position the joints, then right-click ( ) to finish. The bone chain will be created using the type specified in the "Kinematics" option located directly above the viewport. For "Follow Curve" bone chains, expect to be prompted for a curve object. To start drawing from any object, Shift+Left click (Shift) on the object that you wish to extend from.

To change bone chain kinematic types, first choose the "Remove Kinematics" option from the Ctrl+Right mouse (Ctrl) menu. Select the bones that you wish to remove kinematics for, then right-click ( ) to complete. Now, change your Kinematics option to the new type. Finally, choose the "Add Kinematics" option from the Ctrl+Right mouse (Ctrl) menu. Houdini will prompt you for the required object selections to perform the operation.

CHOOSING THE PARENT FOR A NEW BONE CHAIN

To select a parent, Ctrl+Middle (Ctrl) click on the desired object. The new bone chain that you draw after this will have this object as its parent. To reset back to no parent, Ctrl+Middle (Ctrl) click on an empty area in the Viewport.

SPLITING A BONE

To divide a bone into two, Ctrl+Left mouse (Ctrl) click on the desired bone.

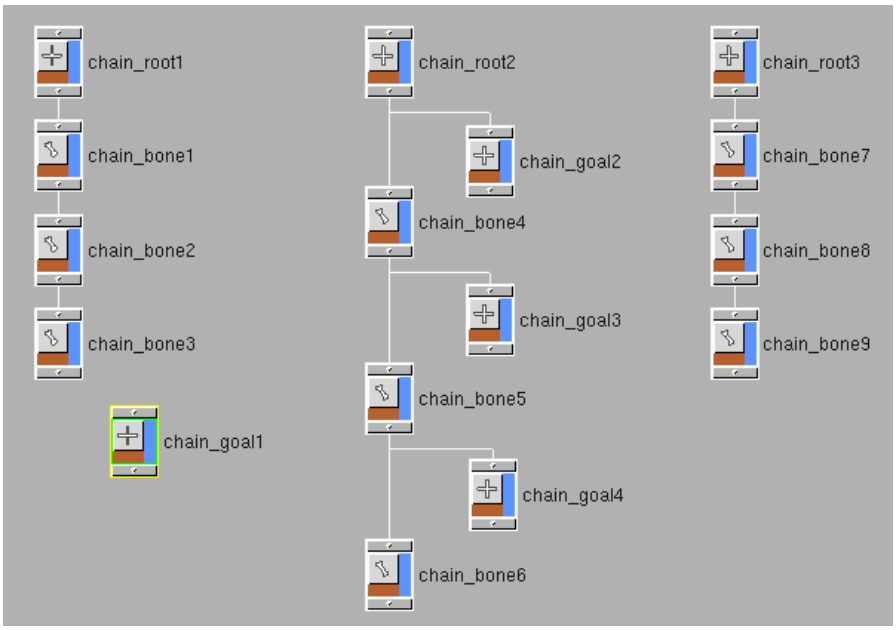
MANIPULATING JOINTS

A Joint handle is created for all the currently displayed bones when you enter the Bones operation. To adjust the position of any joint, click or drag the square handles that you see positioned at the tips of bones. If you do not wish to accidentally click on these squares, toggle them off by pressing the Joint handle button in the left toolbar, or right-click ( ) on any Joint handle and toggle its Display Handle option.

The Joint handle operates in several modes. Right-click ( ) on any of the square handles to see the complete list. The translate handle manipulates the joint position. The rotate handle spins the bone about its origin. The spider handle adjusts the lengths of the bones connected at the joint and moves the joint along the average normal of those bones.

RESULT OF CREATING BONES

After drawing the bones, you will see an assembly of bone objects created in the Layout area. This layout will differ depending on the type of kinematics selected:



Inverse Kinematics *Forward Kinematics* *No Kinematics*

BUILDING CHAINS PARENTED TO OTHER OBJECTS

To add a bone chain that is parented to another object, click with Ctrl in the Viewport to select a parent.

Once you've selected a parent, then build the chain by clicking with . Your first bone will be attached to the parent bone. If you've selected any other type of object then you must click with right-mouse ( ) to finish building the chain.

To select a parent and immediately begin drawing, you can simply Alt click on the parent to save the extra click, and then continue drawing bones with as usual.

To change the parent, either delete all entered bones (with the Del key), or select a new parent from the menu in the Parameters dialog.

If you select a bone as your parent, you will immediately start building attached to the end of the selected bone. If you choose another object type, then you must click again with to start building. The parent will be highlighted while building.

Using Ctrl while drawing will reselect the parent of the bone chain, if chain was started with Shift, then chain will move to the end of the selected bone, or origin of selected non-bone object.

WHAT TO DO NEXT

Before binding the bones to the skin with Capture Geometry, you may want to adjust the capture regions with Edit Capture Regions, then mirror the bone chain with Mirror if you need a symmetrical limb. After Capture, use Pose to animate the bones and see if any capture weights need tweaking. If so, experiment with editing capture regions (procedurally) or with Paint/Edit Capture Weights. Mirror Capture Weights will come in handy if you have painted or hand-edited the weights of one limb and want them reflected automatically to save time.

For a discussion of the other elements in the Character Tools, such as Bone Objects, Bone Chains, and Kinematics, refer to *Character Tools* p. 376.

4.3 OPERATION PARAMETERS

Typing P brings up the full Bones operation settings.

CHAIN NAME

Enter the root name for the created bone chain here.

KINEMATICS MENU

The Kinematics menu has the following types:

- No Kinematics
- Inverse Kinematics
- IK with Twist Affector
- Forware Kinematics
- IK with Constraints
- FK with Constraints
- Follow Curve

See *Character Tools* p. 376 in the *Objects* section for a detailed discussion.

4.4 BONES OPERATION MENU

Invoke the menu for this Operation by using Ctrl RMB.

NO / INVERSE KINEMATIC(S) N / I

Select to create bone chains with *Inverse Kinematics* or *No Kinematics*.

IK WITH TWIST AFFECTOR

Draws bones with twist affector.

FORWARD KINEMATICS F

Draws bones with Forward Kinematics.

IK / FK WITH CONSTRAINTS I / Shift F

Draws bones with full joint constraint while you are creating the bone chain. This allows you to know the effects of the constraint while you draw the bones.

FOLLOW CURVE KINEMATICS Shift K

Draws bones with Follow Curve kinematics.

Tip: When you have a follow curve kinematics bone chain, you can animate the twist of the bones via the *Rest-Z* parameter on the bones.

ADD / REMOVE KINEMATICS

Allows you to specify the start and end of a bone chain, and add / remove a kinematics solution as specified in the Operation Controls *Kinematics* menu.

REMOVE KINEMATICS AND AFFECTORS...

Same as the above, except it will also remove affectors.

TRANSLATE C-PLANE TO END OF LAST BONE T

Moves the origin of the Construction Plane to the end of the last bone.

GO BACK ONE BONE Del

Undoes the last bone added.

FINISH DRAWING

Completes the bone chain.

4.5 PARAMETERS – GENERAL PAGE

CHAIN NAME

Entering a string here to prefixes the names of the created bones as follows:

Entry:	**Creates Names Like:**
if empty	chain_root, chain_goal?, chain_bone?
if filled in	<name>_root, <name>_goal?, <name>_bone?

KIN CHOP NET NAME

Specifies the name of CHOP network which controls the Kinematics of the bone chain.

CHAIN PARENT

Allows you to specify an object to which the bone chain should be parented. If the new parent is a bone, then the entire chain will move to be parented to the new bone, otherwise the chain will not move.

LINK GEOMETRY

Select an object from this menu to define a source for the link geometry of the bones that you draw. The geometry in the specified object will be merged into the bones you create and used instead of the standard diamond-shaped link geometry (For a discussion of link geometry refer to *Bone Object* p. 267 in the *Objects* section.

KINEMATICS

Specify the type of Kinematics to use from the following types:

- No Kinematics
- Inverse Kinematics
- IK with Constraints
- IK with Twist Affector
- Forware Kinematics
- FK with Constraints
- Follow Curve

SET RENDER FLAG ON LINK GEOMETRY

Normally, link geometry is not rendered. This enables the render flag on link geometry.

SET KEEP POSITION WHEN PARENTING FLAG ON NULL OBJECTS

Enables *Keep Position when Parenting* on Null Objects – i.e. when the null object is reparented, its current world position will be maintained.

SET XRAY FLAG ON BONE CHAINS

Sets the XRay flag on created bone chains.

PARENT CHAIN ROOT TO IK AFFECTORS

Determines if you want the end affectors for all IK and FK solvers to be parented to the root of the chain.

4.6 PARAMETERS – EDIT PAGE

Contains options for numerically controlling the position of joints.

EDIT BONE

Select a bone to edit from this menu, the Bone Point will then specify the position of the bone's tip. If you leave it at New Bone if you want to specify a new bone.

If set to *New Bone*, use the *Add / Delete Point* buttons to draw a bone chain; use in the Viewport to finish drawing.

REFERENCE FRAME

Specify if you want the values below to be in reference to obect or world space coordinates. When set to *Object Space*, it will specify a bone point that is relative to the origin of the bone.

BONE POINT

Provides a read-out of the X, Y and Z coordinates of the last clicked point.
Enter values here directly if you want to add a point explicitly.

ADD POINT

By entering the XYZ coordinates of the next point and clicking *Add Point*, you can explicitly enter the bone locations. These are always absolute values – they are specified in world space, not relative to the end of the last bone.

DELETE LAST POINT

Deletes the last point entered. Only works while bones are being drawn.
Clicking this button is equivalent to typing Bksp.

4.7 PARAMETERS – BONE COLORING PAGE

RANDOM CONTRASTING COLORS

When checked, created bones will be assigned random colours that contrast with each other.

SPECIFIC COLOR

When checked, you can assign a specific RGB / HSV colour to the bones you create.

4.8 PARAMETERS – OTHER COLORING PAGE

COLOR CHAIN ROOTS THE SAME AS BONE

When enabled, chain roots will be the same colour as the bones. If not, you can specify a colour using the RGB swatch below.

COLOR AFFECTORS SAME AS BONE

When enabled, bone affectors will be the same colour as the bones. If not, you can specify the Affector colour using the rgb swatch below.

4.9 SEE ALSO

- *Bone Operation* p. 309
- *Edit Capture Region Operation* p. 322
- *Pose Operation* p. 342

5 CAMERA OPERATION

5.1 DESCRIPTION

The Camera operation is used to add a new camera object to your scene. A Camera object defines a viewing location, direction, aspect ratio, and other parameters used when rendering your scene.

Select the object that will be used as the parent for the new Camera object, and right-click ( ) to complete the operation. The new Camera object appears with a handle that can adjust its transform. This transform defines the viewing location and direction. The operation controls bar above the viewport also provides access to the camera's Translate and Rotate parameters.

You can change the Camera object's parent object any time by hitting the Reselect Geometry button or typing . If Secure Selection is turned off, selecting a new object will immediately complete the selection and connect the new parent. Otherwise, you right-click ( ) to signify that the object selection is complete.

If Secure Selection is turned off, you can select any other object in the viewport. Doing so will switch to the operation appropriate for that object type, and the current operation will be exited immediately.

5.2 SEE ALSO

- *Camera Object* p. 271

6 CAPTURE GEOMETRY OPERATION

6.1 DESCRIPTION

The Capture Geometry operation lets you capture one or more geometry objects as the skin for a hierarchy of bones.

The first prompt is to select the geometry objects to capture. In the second prompt, you select the root object of your bones (the skeleton). This does not have to be a bone but any object that is a parent of all the bones that you want captured. Typically, this is a Null object that is an ancestor of your entire skeleton. If the root is hard to pick in the viewport, you may find it easier to select in the network pane.

The geometry points will be colored to show the weighting contribution of the captured bones, if any. Points colored white after the capture have not been captured by any bones. The color contribution of each point is based upon the object wire color of its captured bones.

6.2 CAPTURE METHOD PARAMETER

The Capture Method parameter specifies the method of capturing. The Regions method will capture points using the capture regions of each bone to decide which point is captured. This method may not capture all points initially if some points are outside of all capture regions in the skeleton hierarchy. You can interactively manipulate the size of the regions at any time using the Edit Capture Regions operation. The advantage of this method is that it will allow you to also deform using the capture regions later on when animating.

The Proximity method will usually capture all points of your geometry based on the closest 2 bones. After using the Proximity method, you can drag the middle mouse button in the viewport to adjust the global drop off effect of the bones. Note that you still may not capture all points if you lower the drop off too much. For best results, you will want to adjust your drop off until it just captures all your points. The advantage of this method is that there is only one control (the dropoff) for the entire skeleton.

To fully understand what is done behind the scenes for these two methods, you may wish to look at the SOPs that are added for each of the captured geometry objects.

6.3 CHANNEL GROUP PREFIX AND CAPTURE FRAME PARAMETERS

The Capture Frame parameter specifies the frame number at which to perform the capturing. Keyframes will be created at this frame on the bone rotation and end affector transform parameters of the skeleton. This ensures that your capturing information will not change if you accidentally adjust a non-animated parameter that modifies the capture frame.

The Channel Group Prefix specifies the string prefix of the channel groups that are created to hold these newly created channels.

6.4 WHAT TO DO NEXT

After capturing the geometry, use Pose to animate the bones and see if any capture weights need tweaking. If so, experiment with editing capture regions (procedurally) or with Paint/Edit Capture Weights. Mirror Capture Weights will come in handy if you have painted or hand-edited the weights of one limb and want them reflected automatically to save time.

6.5 SEE ALSO

- *Edit Capture Region Operation* p. 322
- *Edit Capture Weight Operation* p. 323
- *Paint Capture Weights* p. 338
- *Mirror Capture Operation* p. 335
- *Pose Operation* p. 342

7 COPY OPERATION

7.1 DESCRIPTION

The Copy operation allows you to make copies objects within the same subnetwork.

It features automatic copying of the associated InverseKin CHOPs and affector objects of bone chains.

For the full set of options, please see the Operation Parameters dialog via the P key.

Note: Not to be confused with the Copy Operation (SOP) available in *Geometry.*

7.2 PARAMETERS

NEW PREFIX

The prefix to add to the name of the copied objects

REMOVE OLD PREFIX

Turn on removal of old prefixes in the copied names

end character

Specifies the end character of the old prefix to be removed.
Everything up to and including this character will be removed.

NEW SUFFIX

The suffix to add to the name of the copied objects

REMOVE OLD SUFFIX

Turn on removal of old suffixes in the copied names

start character

Specifies the start character of the old suffix to be removed.
Everything after this character and itself will be removed.

COPY INVERSEKIN CHOPS...

Copy Inversekin CHOPs when Copying Whole Bone Chains will enable the automatic copying of associated InverseKin CHOPs.

TARGET CHOP NET

Specifies the network name to copy the InverseKin CHOPs to. If left blank, they will be copied to the same network as their originals.

8 EDIT CAPTURE REGION OPERATION

8.1 DESCRIPTION

The Edit Capture Region operation allows you to adjust the bone capture regions.

After selecting your bones and right-clicking ( ), you are presented with a capture region handle for each bone in your selection. These handles manipulate the parameters found in the Capture page of the Bone parameters dialog.

To adjust the region center, use the provided translate handle. Dragging on the arrows of the capture region handle will adjust the top or bottom cap radius. To adjust both the X and Y radius simultaneously, drag on the cross arches of the caps along the length of the bone. To adjust the top and bottom heights, drag on the circles along the length of the bone. Finally, you can symmetrically adjust both the top and bottom portions at the same time by holding down the Shift key while dragging.

This handle can be also be used as a deformer by keyframing them on captured geometry. To do this, you MUST first make sure that you create a keyframe at the capture frame so that you do not modify any of the capture weights. An easy method to do this is to first go to the capture frame on the playbar. The default is frame 0 in the Capture Geometry operation. Now create handles on all the bones you wish to animate. If you wish to select all bones, you can use the A key. After right-clicking ( ) to create the handles, use the pull-down menu that is located next to the toolbar near the top of the Viewport. Choose the "Set Keyframes For All Handles" option. This will create all the necessary keyframes. Now to animate, move the playbar to the frame you wish to change and right-click ( ) on the handle to do a "Set Keyframe". You can now adjust the handle to deform your geometry. Make sure you also turn off the "Only Update Capture" operation parameter.

You can edit the capture regions before or after capturing the geometry. Editing the regions before Capture has the advantage that if you plan to mirror the bones whose regions you are editing, the changes you are making at this stage will automatically be inherited by the mirrored bones once you call up Mirror. Editing the regions after Capture has the advantage that you can instantly watch the geometry shape and color change as you modify the capture regions.

8.2 SEE ALSO

- *Capture Geometry Operation* p. 319
- *Edit Capture Weight Operation* p. 323
- *Paint Capture Weights* p. 338
- *Mirror Capture Operation* p. 335

9 EDIT CAPTURE WEIGHT OPERATION

9.1 DESCRIPTION

The Edit Capture Weights operation is used to individually modify point capture weights using slider handles on bones and/or via a spreadsheet.

After selecting the captured points that you wish to modify, slider handles are created for all currently displayed bones. The position of the slider knobs on the bones indicates its average influence in the point selection. The closer the knob is to the tip of its bone, the higher the influence. To change your point selection, use the "Reselect geometry" button in the left toolbar at any time.

When the slider knob is adjusted, the influence of the bone is added relatively to the point selection if it is moved towards the tip of the bone, and subtracted when moving away. To use absolute weight, hold down the shift key while adjusting the knob. This will change the weight of that bone in all the points to be the same. You can also numerically control the slider by middle-clicking ( ) and then dragging on the knob.

If you right-click ( ) on the slider, you have some options controlling the slider. Of note, the "Clamp negative weights" option is on by default. This means that when subtracting weight from your point selection, it will never result in negative weights, stopping at a weight of 0. The "Absolute Mode" option makes the default dragging behaviour to be in absolute terms as opposed to relative terms. This option can be thought of as always holding down the Shift key while adjusting the slider.

Note: When in *Absolute Mode*, the *Clamp Negative Weights* option in the slider has no effect.

The weighted bones are always highlighted after making a point selection. This indicates that they are available in the spreadsheet. You can bring up the spreadsheet by pressing the spreadsheet button in the operation controls bar directly above the viewport. To add or remove capture regions from the spreadsheet, Ctrl+Middle mouse (Ctrl) click on the bone in the Viewport. To change a selection here to be all the same value, highlight the cells and type the weight in the active cell of the highlighted selection. If you use the middle mouse to drag in this active cell, then the weights will be changed relatively instead. To have your cell selection to be highlighted in the viewport instead, change the Manipulate option in the operation dialog "Spreadsheet Selection".

For more options, see the Ctrl+Right (Ctrl) mouse button menu in the Viewport.

9.2 SEE ALSO

- *Capture Geometry Operation* p. 319
- *Edit Capture Region Operation* p. 322
- *Paint Capture Weights* p. 338
- *Mirror Capture Operation* p. 335

10 FETCH OPERATION

10.1 DESCRIPTION

The Fetch operation is used to fetch a transform from another object in the scene. It is used to extract a transform that is inside a different object subnet, or to simply duplicate an existing object's transform.

You select the object that will be the parent for the new Fetch object, and right-click ( ) to complete the operation. The new fetch object appeats, but there are no handles for this operation.

The Fetch object can fetch the local transform of the object, in which case the output of the Fetch object is the world transform of its parent object multiplied by the local transform of the fetched object. It can also fetch the world transform, in which case the Fetch object creates a local transform such that its parent object's world transform multiplied by its own local transform is equal to the world transform of the object being fetched. This mode of operation is controlled by the Fetch World Transform parameter available in the operations controls bar above the viewport. The controls bar also contains a menu to select the object from which to fetch the local or world transform.

You can change the Fetch object's parent object any time by hitting the Reselect Geometry button or typing . If Secure Selection is turned off, selecting a new object will immediately complete the selection and connect the new parent. Otherwise, you right-click ( ) to signify that the object selection is complete.

If Secure Selection is turned off, you can select any other object in the viewport. Doing so will switch to the operation appropriate for that object type, and the current operation will be exited immediately.

10.2 SEE ALSO

- *Blend Operation* p. 308

11 FOLLOW PATH OPERATION

11.1 DESCRIPTION

This Operation simply sets the Path Object parameter within a Geometry Object.

Use to select the object you want to be on the path (and to complete selection), and then click to pick the path object you want it to be on (and to complete) – the Path object for the object will be set (you can view its parameters > *Transform* page to see this).

Note: You will still have to enter the object to animate its position along the path. You could do this by setting the Position parameter to something like:

```
Position: $F/$NFRAMES
```

11.2 SEE ALSO

- *Geometry Object* p. 280

12 FOG OPERATION

12.1 DESCRIPTION

The Atmosphere operation is used to create atmospheric effects in your scene that appear when the scene is rendered.

You select the object that will be used as the parent for the new Atmosphere object, and right-click ( ) to complete the operation.

You can change the Atmosphere object's parent object any time by hitting the Reselect Geometry button or typing . If Secure Selection is turned off, selecting a new object will immediately complete the selection and connect the new parent. Otherwise, you right-click ( ) to signify that the object selection is complete.

If Secure Selection is turned off, you can select any other object in the viewport. Doing so will switch to the operation appropriate for that object type, and the current operation will be exited immediately.

13 FORCE OPERATION

13.1 DESCRIPTION

The Force operation is used to create an object level representation of a force that can be referenced by a Force POP. It can also be used to represent a constraint force used by the Constraint POP. The most common use for this object is to provide control for Rigid Body Dynamics simulations.

You select the object that will be used as the parent for the new Force object, and right-click ( ) to complete the operation. The new Force object appears with a transform handle to modify the direction and magnitude of the force. The rotate and scale parameters are also available in the operation controls bar above the viewport.

You can change the Force object's parent object any time by hitting the Reselect Geometry button or typing . If Secure Selection is turned off, selecting a new object will immediately complete the selection and connect the new parent. Otherwise, you right-click ( ) to signify that the object selection is complete.

If Secure Selection is turned off, you can select any other object in the viewport. Doing so will switch to the operation appropriate for that object type, and the current operation will be exited immediately.

13.2 SEE ALSO

- *Rigid Body Dynamics Operation* p. 345

14 GEO OPERATION

14.1 DESCRIPTION

The Geometry operation is used to create a new piece of geometry for your scene. Use this operation to create the objects in your scene that will appear when rendering your scene.

You select the object that will be used as the parent for the new Geometry object, and right-click ( ) to complete the operation. The new Geometry object appears with a transform handle that can be used to change the position and orientation of the geometry. The translate and rotate parameters also appear in the operation controls bar above the viewport.

To edit the geometry itself, select Model Current Object from the viewport menu. Or you can right-click ( ) on any object, and choose Model This Object from the popup menu.

You can change the Geometry object's parent object any time by hitting the Reselect Geometry button or typing . If Secure Selection is turned off, selecting a new object will immediately complete the selection and connect the new parent. Otherwise, you right-click ( ) to signify that the object selection is complete.

If Secure Selection is turned off, you can select any other object in the viewport. Doing so will switch to the operation appropriate for that object type, and the current operation will be exited immediately.

15 HANDLE OPERATION

15.1 DESCRIPTION

The Handle operation is used to create a new Handle object. Handle objects are used to control the motion of bones using motion capture data. Each Handle object represents a control point in the motion capture data, and can influence the motion of a particular bone object. Generally, you will use one of the motion capture conversion applications to generate a script that creates these objects. You will not generally create objects of this type manually.

You select the object that will be used as the parent for the new Handle object, and right-click ( ) to complete the operation. There are no handles associated with this operation.

The operation controls bar above the viewport provides access to the Target and Translate parameters of the Handle object.

You can change the Handle object's parent object any time by hitting the Reselect Geometry button or typing . If Secure Selection is turned off, selecting a new object will immediately complete the selection and connect the new parent. Otherwise, you right-click ( ) to signify that the object selection is complete.

If Secure Selection is turned off, you can select any other object in the viewport. Doing so will switch to the operation appropriate for that object type, and the current operation will be exited immediately.

16 LIGHT OPERATION

16.1 DESCRIPTION

The Light operation is used to create a new directed light source in your scene. Light objects have a position and direction associated with the light they emit. They can also project textures, produce a cone of light, and produce a number of other effects.

You select the object that will be used as the parent for the new Light object, and right-click ( ) to complete the operation. The new Light object appears with a transform handle that can be used to change the position and orientation of the light. The light color and dimmer parameters also appear in the operation controls bar above the viewport.

You can change the Light object's parent object any time by hitting the Reselect Geometry button or typing . If Secure Selection is turned off, selecting a new object will immediately complete the selection and connect the new parent. Otherwise, you right-click ( ) to signify that the object selection is complete.

If Secure Selection is turned off, you can select any other object in the viewport. Doing so will switch to the operation appropriate for that object type, and the current operation will be exited immediately.

17 LOOK AT OPERATION

17.1 DESCRIPTION

This Operation sets the *Look At* parameter within a Geometry Object.

Use to select the object you want to Orient (and to complete selection), and then click to pick the object you want it to Orient Towards (and to complete) – the *Look At* parameter for the object will now be set for you (you can view its parameters > *Transform* page to see this).

17.2 SEE ALSO

- *Geometry Object* p. 280

18 MATERIAL OPERATION

18.1 DESCRIPTION

The Material operation allows you to assign materials to objects interactively.

First select one or more objects to apply material to, then right-click ( ) to assign the material specified in the menu immediately above the Viewport.

Note: Not to be confused with the geometry-level Material Operation (SOP)

18.2 PARAMETERS

WORK WITH SHADERS

Enables working with SHOPs (instead of Materials / TOPs).

MATERIAL

Pick the material you want to assign from this menu.

19 MICROPHONE OPERATION

19.1 DESCRIPTION

The Microphone object defines a listening point for the SpatialAudio CHOP. Multiple microphones can be used by one SpatialAudio CHOP to create stereo or surround sound, with special effects like the doppler effect, volume loss over distance, obstacle interference, atmospheric filtering and positional audio.

To set up a Spatial Audio scene, one or more Sound objects should be used to emit sound. At least one microphone is needed to capture the sound. A SpatialAudio CHOP is needed to render the sound. If any obstacles or filters are used, at least one Acoustic CHOP is needed to design the spectrum filter.

Moving Sound and Microphone objects around will produce variations in pitch and volume, especially if either object is directional. Setting up a directional microphone or sound object is much like setting up a directional light.

You select the object that will be used as the parent for the new Microphone object, and right-click ( ) to complete the operation. The new Microphone object appears with a transform handle that can be used to change the position and orientation of the microphone. The sensitivity, directional toggle and recording cone parameters appear in the operation controls bar above the viewport.

You can change the Microphones's parent object any time by hitting the Reselect Geometry button or typing . If Secure Selection is turned off, selecting a new object will immediately complete the selection and connect the new parent. Otherwise, you right-click ( ) to signify that the object selection is complete.

If Secure Selection is turned off, you can select any other object in the viewport. Doing so will switch to the operation appropriate for that object type, and the current operation will be exited immediately.

19.2 SEE ALSO

- *Sound Operation* p. 346
- *Light Operation* p. 330

20 MIRROR OPERATION

20.1 DESCRIPTION

The Mirror operation allows you to mirror individual objects or entire hierarchies.

It will mirror objects along one of the XY/YZ/ZX planes of the eldest parent object in your selection. For example, say that you had a shoulder and some arm bones that you wished to mirror. Furthermore, the shoulder bone was a child of a spine bone. To mirror these bones, you would select them when prompted in the status line located next to the main Houdini menu. Since the highest object in the hierarchy of the selection is your shoulder bone, the mirror plane will reside in the shoulder bone's parent (a spine bone). The Mirror operation will mirror about the eldest parent for each cluster of hierarchically connected objects in the selection.

As soon as you select any set of objects, the mirrored selection will automatically appear as templated geometry without creating the mirror objects yet. You can adjust the mirror plane and preview the results before right-clicking ( ) to complete the operation.

The "New prefix" option specifies the prefix to add to the new names of the mirrored objects. The "Remove old prefix up to" option specifies the end character that is removed from the old name up to and including itself. For example, if you mirrored an object named "left_shoulder" you might want the new prefix to be "right_".

20.2 SEE ALSO

- *Mirror Capture Operation* p. 335

21 MIRROR CAPTURE OPERATION

21.1 DESCRIPTION

The Mirror Capture Weights operation provides a special method for modifying the capture weights of a geometry object. It allows you to take the capture weights from one set of points and bones and mirror them through an arbitrary plane onto other parts of the geometry.

To perform this operation, you first select the points in your geometry that you wish to mirror. Only points from a single object can be chosen, and the chosen object must either have capture weight attributes, or it must have a Deform SOP, with capture weight attributes defined above the Deform SOP. Once the points are chosen, right-click ( ) to proceed.

The second step in this operation is to select the bone objects that contain the source capture regions. When you complete the selection of your points, all bones with capture regions with non-zero weights on those points are selected automatically. You can either use this default selection, or modify it, then right-click ( ) to proceed. Only objects that contain capture regions can be included in this selection.

The third step is to choose the bones that contain the destination capture regions. When you complete the selection of your source bones, you will again be presented with a default selection. This default selection is found by trying to find a set of bones that match the names of the source bones. It tries to replace the prefix of the source bones with a different prefix to generate the destination bone names. It finds the shortest prefix possible to allow this substitution. Again, you can either accept this default selection, or odify the selection. Right-click ( ) to complete the operation.

A Capture Mirror SOP will be created in the object where you are mirroring capture weights. The SOP will be created as the input to the Deform SOP if there is one, or as the end of the SOP chain if there is no Deform SOP. You will be presented with handles that allow you to define the plane through which the capture weights are mirrored.

21.2 SEE ALSO

- *Capture Geometry Operation* p. 319
- *Mirror Operation* p. 334
- *Edit Capture Weight Operation* p. 323
- *Edit Capture Region Operation* p. 322
- *Paint Capture Weights* p. 338

22 NULL OPERATION

22.1 DESCRIPTION

The Null operation is used to create an object that is most often used to provide control over the transforms of other objects. Null objects are not rendered, but they have full transforms and so can be used as parents for other objects, or serve as the end affector for Inverse Kinematics bone chains.

You select the object that will be used as the parent for the new Null object, and right-click ( ) to complete the operation. The new Null object appears with a transform handle that can be used to change its position and orientation. The translate and rotate parameters also appear in the operation controls bar above the viewport.

You can change the Null object's parent object any time by hitting the Reselect Geometry button or typing . If Secure Selection is turned off, selecting a new object will immediately complete the selection and connect the new parent. Otherwise, you right-click ( ) to signify that the object selection is complete.

If Secure Selection is turned off, you can select any other object in the viewport. Doing so will switch to the operation appropriate for that object type, and the current operation will be exited immediately.

23 OBJECTS OPERATION

23.1 DESCRIPTION

The Objects operation is used to manipulate the handles of several objects at the same time. By providing one handle for each selected object, this operation allows more flexiblity than the Transform operation which provides a single handle to manipulate all selected objects.

Select the objects you want to manipulate, and right-click ( ) to complete the operation. You will be presented with handles for each object in the selection. The handles are created based on the manipulator bindings defined for ech object. Most object have a transform handle, but some (such as the Bone object) have handles more specifically suited to the object.

If Secure Selection is turned off, you can select any other object or group of objects in the viewport. Handles will immediately appear for this new set of objects. You can also select objects in a network editor, and the handles will appear for this new selection.

23.2 SEE ALSO

- *Transform Operation* p. 349

24 PAINT CAPTURE WEIGHTS

24.1 DESCRIPTION

The Paint Capture Weights operation allows you to interactively paint the capture weights of your geometry.

After choosing this operation, your first task is to select the geometry to paint, then right-click ( ) to complete. The geometry must all be within one object node.

As you move the mouse over the geometry, you will see a circular brush follow the surface to show where you would paint. The size of the brush and its orientation will show you which points will be affected by a paint operation.

Painting is done to the weights of one bone at a time. You may change which bone by Ctrl-middle clicking the desired bone.

Like all the brush tools, you have two different brushes: your Foreground brush and your Background brush. You paint with your foreground brush using the and the background with the . The amount of weight painted is controlled by the "FW" and "BW" parameters respectively.

Note: Refer to Interface > *Brush Tools* p. 139 ***for useful info!***

24.2 TOOLS

OPERATION MENU (Ctrl)

Each of the brushes also can have its own independent operation. To select the operation for each brush, use Ctrl mouse to bring up the operation menu. The options are:

PAINT

This paints your new weight onto the geometry.

EYE DROPPER

This takes the weight of the point nearest the center of your brush and sets it to that brush's painting weight.

SMOOTH

This smoothly blends the weights of nearby points under the brush.

ERASE CHANGES

This is a local undo - it restores the points under your brush to the pre-capturepaint values.

24.3 OTHER PARAMETERS

The other important parameters are:

VISUALIZE

This controls how to display the bone's weightings. The default "bone color" will color each point by the amount each bone affects it with that bones color. This is useful to see what bone controls which point. The "false color" will only show that bone's weights, but allow you to tell with greater precision how much any point is affected by the bone.

RADIUS

The size of the brush.

OPACITY

How strong the brush is. A fully opaque brush replaces the original weight, while a less opaque brush blends it.

24.4 OPERATION MENU (Ctrl)

The other Ctrl-right mouse menu options are:

CIRCLE / SQUARE / BITMAP BRUSH

The shape of the brush. The bitmap is set in the operation parameters dialog.

ORIENT BRUSH TO SURFACE

When disabled, the brush is always oriented the same way. This is useful when painting crinkly geometry.

ACCUMULATE TO STENCIL

One of the more powerful modes to work with in the Brush Operations is the *Accumulate To Stencil* mode. Unfortunately, it isn't as intuitive as the normal, immediate application, mode. To understand accumulate to stencil, it is important to note that when the mode is off, you are still accumulating to a stencil. You are just automatically applying it whenever you let go of the mouse button so remain unaware of the fact. However, some operations are significantly easier with this mode on.

For an example, see: Interface > *Accumulate To Stencil Demystified* p. 144.

24.5 SEE ALSO

- *Capture Geometry Operation* p. 319
- *Edit Capture Weight Operation* p. 323
- *Edit Capture Region Operation* p. 322
- *Mirror Capture Operation* p. 335

25 PARENT OPERATION

25.1 DESCRIPTION

This Operation sets the Parenting relationship between Objects (this is normally accomplished by wiring Object nodes together in an Objects *Network Editor*.

Use to select the Child you want to Parent (and to complete selection), and then click to pick the object you want to Parent it to (and to complete) – the wiring connection between these objects in the Network Editor will now be set for you.

Note: You may want to set the *Keep Position when Parenting* option to keep your geometry from moving around when you parent it.

26 POSE OPERATION

26.1 DESCRIPTION

The Pose operation is used to pose characters by manipulating the Bone objects. The bones can be posed whether they are using an Inverse Kinematics solver, a Forward Kinematics solver, or no kinematics solver.

Select the bone objects to maipulate, and right-click ( ) to complete the operation. You will be presented with one or more Pose handles. Each separate bone chain will be given its own Pose handle. Bone chains that branch will have a separate Pose handle for each branch. Any bones that are controlled by an Inverse Kinematics solver will have a Pose handle attached to all the bones in that chain (even if not all of the bones were selected). These handles let you manipulate the bones in a variety of ways, depending on the type of solver applied to the bone chain.

For bone chains with an Inverse Kinematics solver, you can move the end affector and you can apply a twist. The twist will move the twist affector if there is one; otherwise, it will modify the Twist parameter of the solver.

For bone chains with Forward Kinematics or no kinematics, you can select any bone in the chain and manipulate it directly. You can also select the transform handle at the end of any joint, and manipulate the handle. Doing so will use inverse kinematics to assign new rotations to evey bone higher up the chain. The transform handle is treated as if it were an end affector for the chain. For chains using Forward Kinematics, the bones are manipulated by transforming the end affector for each bone. For chains with no kinematics, the bones are manipulated by changing the Rotate parameter of the Bone objects.

Any joint in the chain can also be locked by shift-left-clicking on the transform handle at that joint. Locked joints appear with their transform handles solid shaded. Unlocked joints appear with their transform handles in wireframe. When a joint is locked, modifications made higher up the bone chain are not allowed to affect the position of the locked joint.

If Secure Selection is turned off, you can select any group of bones in the viewport. Pose handles will immediately appear for this new set of bones. You can also select bones in a network editor, and the Pose handles will appear for this new selection.

WHAT TO DO NEXT

When posing the character, see if any capture weights need tweaking. If so, experiment with editing capture regions (the procedural approach) or with Paint/Edit Capture Weights. Mirror Capture Weights will come in handy if you have painted or hand-edited the weights of one limb and want them reflected automatically to save time.

26.2 SEE ALSO

- *Bones Operation* p. 310
- *Objects Operation* p. 337
- *Transform Operation* p. 349

27 RENDER OPERATION

27.1 DESCRIPTION

The Render operation allows you to visualize the rendered scene in a portion of the viewport in real time.

Choose a valid output driver from the menu immediately above the viewport or use the default mantra renderer, then box-select the area of the screen you want rendered.

The rendered image will update automatically as you make changes to the scene elements or to the viewing position. You can also force a new rendering by clicking the "Re-Render" button above the viewport.

The default renderer will not render with motion blur or depth of field. To render with these effects, please choose an output driver which has the requisite options specified.

No shadow/reflection maps will be auto-generated by the Render Operation. All maps must be pre-rendered for correct shading to occur.

Note: Changes to geometry are not automatically detected (including changes to indirect SHOP references), when geometry changes, it may be necessary to hit the *Re-Render* button.

28 RIGID BODY DYNAMICS OPERATION

28.1 DESCRIPTION

The Rigid Body Dynamics operation is used to aid in setting up a POP network to create a RBD simulation.Like the Bones Operation, everything done here can be manually achieved (by wiring up networks of POPs and CHOPs). This Operation just simplifies their setup. From your specifications within this Operation, it creates a POP network, and possibly a number of Force Objects according to you selections and commands. The Force Object is a way of creating a visual representation in world space of the parameters of certain POPs (Force, Fan, and Constraint).

To start this operation, select the geometry objects that you wish to have included in the RBD simulation, and right-click ( ) to complete the operation. Once you have chosen the objects that will be in the simulation, this operation gives you several ways to modify the simulation.

The Ctrl-Right-mouse (Ctrl) menu lets you create a force, a fan, or a nail, point, or rotation constraint. Each of these menu items create a Force object and a POP that corresponds to the menu item you chose. The parameters of the Force object are also set to create the desired type of force.

The operation controls bar above the viewport allows you to turn on Grouping mode. When in this mode, it is easiest to have Secure Selection turned off. However, you can always accomplish the same tasks by having Secure Selection turned on and right-clicking ( ) after performing each selection. This mode allows you to pick which objects in the simulation are affected by which Force objects. You begin by selecting the Force object that you wish to do grouping for. Then you select the object or objects that you want that Force to apply to.

The Point constraint force acts on two sets of objects, constraining each object in the first group to the corresponding object in the second group. To do this in the viewport using Grouping mode, first you select the point constraint object, and then select the objects for the first group. Then turn on the Group B parameter in the operation controls bar. Then select the point constraint object again, and then select the objects for the second group of the point constraint.

29 SOUND OPERATION

29.1 DESCRIPTION

The Sound object defines a sound emission point for the SpatialAudio CHOP. Multiple sound sources can be mixed by one SpatialAudio CHOP to create stereo or surround sound, with special effects like the doppler effect, volume loss over distance, obstacle interference, atmospheric filtering and positional audio.

To set up a Spatial Audio scene, one or more Sound objects should be used to emit sound. At least one microphone is needed to capture the sound. A SpatialAudio CHOP is needed to render the sound. If any obstacles or filters are used, at least one Acoustic CHOP is needed to design the spectrum filter.

Moving Sound and Microphone objects around will produce variations in pitch and volume, especially if either object is directional. Setting up a directional microphone or sound object is much like setting up a directional light.

You select the object that will be used as the parent for the new Sound object, and right-click ( ) to complete the operation. The new Sound object appears with a transform handle that can be used to change the position and orientation of the sound source. The volume and CHOP audio source parameters appear in the operation controls bar above the viewport.

You can change the Sound object's parent object any time by hitting the Reselect Geometry button or typing . If Secure Selection is turned off, selecting a new object will immediately complete the selection and connect the new parent. Otherwise, you right-click ( ) to signify that the object selection is complete.

If Secure Selection is turned off, you can select any other object in the viewport. Doing so will switch to the operation appropriate for that object type, and the current operation will be exited immediately.

29.2 SEE ALSO

- *Microphone Operation* p. 333
- *Light Operation* p. 330

30 SUBNET OPERATION

30.1 DESCRIPTION

The Subnet operation is used to create a new logical grouping of objects. Subnet objects contain other objects, not SOPs like most object types. This lets you organize your object hierarchy.

You select the object that will be used as the parent for the new Subnet object, and right-click ( ) to complete the operation. There are no handles defined for this operation.

The operation controls bar above the viewport provides access to the Display parameter of the Subnet. When the Display parameter for a Subnet is set to 0, none of the objects contained in the Subnet are displayed. When this parameter is set to 1, the Display parameter of the contained objects are used to determine the visibility of each object. The operation controls bar also lets you specify which object\'s transform should be used as the output from the Subnet. You can either choose to pass through the transform from one of the Subnet inputs, or to output the transform of one of the contained objects.

You can change the Subnet object's parent objects any time by hitting the Reselect Geometry button or typing . If Secure Selection is turned off, selecting new objects will immediately complete the selection and connect the new parents. Otherwise, you right-click ( ) to signify that the object selection is complete.

If Secure Selection is turned off, you can select any other object in the viewport. Doing so will switch to the operation appropriate for that object type, and the current operation will be exited immediately.

31 SWITCHER OPERATION

31.1 DESCRIPTION

The Switcher operation is used to create a new object that allows you to switch easily between several cameras.

You select the cameras that you wish to switch between. Once the objects are selected, right-click ( ) to complete the operation. There are no handles associated with this operation.

The operation controls bar above the viewport contains a parameter which lets you switch between the various input cameras. The Switcher object gets all its viewing parameters from the currently selected input camera.

You can specify a different set of parent cameras by hitting the Reselect Geometry button or typing . If Secure Selection is turned off, selecting one or more new cameras will immediately complete the selection and connect the new parents. Otherwise, you right-click ( ) to signify that the selection is complete.

If Secure Selection is turned off, you can select any other object in the viewport. Doing so will switch to the operation appropriate for that object type. The current operation will be exited immediately.

31.2 SEE ALSO

- *Camera Operation* p. 318

32 TRANSFORM OPERATION

32.1 DESCRIPTION

The Transform operation is used to manipulate the object transforms. You can manipulate either single objects or groups of objects using a single transform handle.

You select the objects to transform, and right-click ( ) to complete the operation. A single transform handle will appear. Manipulating this transform handle will change the transforms of all selected objects. Selecting objects while in this operation also causes the object Pick Script to be executed. Pick Scripts can run any series of Houdini script commands, and can even modify the current selection.

If multiple objects are selected, and one selected object is the parent of another selected object, the transform handle will recognize this fact. The object transforms will be set such that the group of selected objects will behave as if it were a single piece of geometry. From the right-mouse menu on the transform handle, you can enable Ignore Parenting. Doing so will cause the transform handle to ignore the parenting of the selected geometry (his mimics the behaviour of the Transform handle in Houdini 4).

The operations controls bar allows you to turn on Link to Channel Editor and Link to CHOP Editor. If these features are enabled, any animated parameters of the selected objects will be scoped in any open channel editor or CHOP editor window.

If Secure Selection is turned off, you can select any other object or group of objects in the viewport. The transform handle will immediately become attached to this new selection of objects. You can also select objects in a network editor, and the transform handle will attach to this new selection of objects.

32.2 SEE ALSO

- *Objects Operation* p. 337

33 VIEW OPERATION

33.1 DESCRIPTION

The View operation lets you change the viewing position within the Viewport.

Note: Not to be confused with the Geometry View Operation (SOP).

MOUSE CONTROLS

LMB	Tumble
MMB	Dolly
RMB	Pan
Alt LMB	Zoom
Ctrl LMB	Rotate
Ctrl MMB	Box Zoom
Ctrl RMB	Menu

KEYBOARD SHORTCUTS

Space	Volatile key – temporarily enter this state from any other Operation
Esc	Return to this Operation from any other.
Ctrl 1 - 9	Change the Layout of the Viewport panes.
H	Home on the visible geometry.
Shift H or G	Home on selected geometry.
V	Save current view to Camera if viewing through one.
W	Toggle between Shaded and Wireframe mode.
D	Display Options Dialog.

NOTES

If the *Lock Camera to View* option is off, changing the viewing position when looking through a camera will set the camera menu to "*no camera*". If this option is on, changing the view will adjust the camera.

Multiple viewing positions can be saved and restored via view memories. To access these memories, click on the stow bar at the bottom of the Viewport.

4 Dialog Scripts For Objects

1 DIALOG SCRIPTS

1.1 COMMONALITIES

SELECTION MENU

Allows you to specify an item whose parameters you wish to modify.

RELOAD

The *Reload* command is used when you have edited a dialog script in *$HH/config/scripts* and wish to see the changes reflected immediately in the dialog.

HELP

Displays the contextual help for the item whose parameters you are editing.

PRESETS

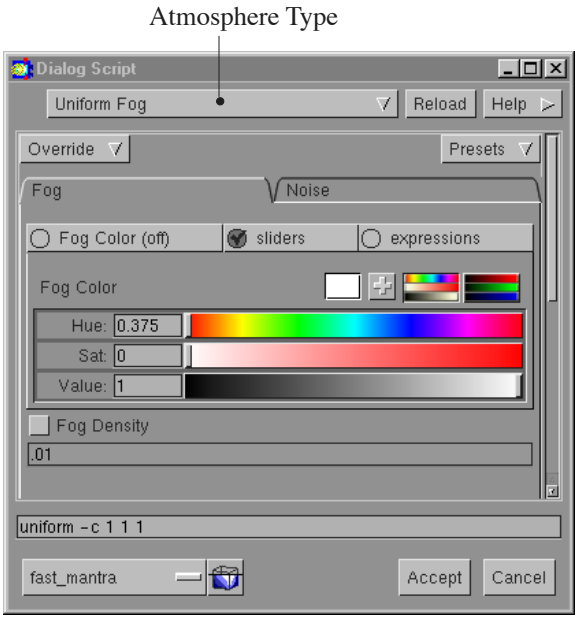
The *Presets* menu allows you to: *Load* parameters used previously for this dialog; *Save* the current session's parameters; and return to the dialog's *Default* settings.

Render icon and Output Driver menu

TEST RENDERING CONTROLS

This menu and icon allow you to select an output driver with which to perform a test render without having to leave the Dialog Script.

2 ATMOSPHERE TYPES DIALOG



LOCATION OF DIALOG

Atmosphere object > *Shading* page > *Shader* ✣ parameter

2.1 ATMOSPHERE TYPE MENU

LIGHT HALO

Light Halo fog is used for environment lights that have an atmosphere variable specified. The Ramp parameter accesses a color ramp. This ramp color will be applied to the halo radially outward. Referencing the ramp depends on the intensity of the halo fog at that point. In addition to color, density, and ramp you can set the attenuation parameter for this atmosphere type.

LENS FLARE

Houdini's implementation of the lens flare effect is three dimensional. Because it is implemented through the Atmosphere object in world space, the flare object can take on any shape you choose, and can be animated–with occlusion being, in part, a function of where the object travels in world space.

LAYERED FOG

Layered Fog generates a stratified atmosphere. The *min./max. height* variable lets you specify the height of the layers. The *Ramp* parameter lets you specify a color ramp to use for the fog. The color of the fog will be multiplied by the value contained in the ramp. *Near/Far* allows you to vary the density of the fog with distance.

UNIFORM FOG

Uniform Fog creates an atmosphere with uniform density. You can modify the color of the atmosphere by entering red, green, and blue values in the edit fields. You can also specify the fog density. A value of one creates a very thick fog.

Z-DEPTH FOG

Z Depth Fog considers depth cuing and will more accurately render objects at different positions in the Z axis.

Note: Transformations affect the Light Halo and Layered Fog atmosphere types. All others are transform independent.

2.2 PARAMETERS – FOG PAGE

OFF / SLIDERS / EXPRESSIONS

Turns colours off, set them using regular sliders, or define the colours using expressions to define the RGB values.

FOG COLOR

Sets a tint color for the fog. When the fog is generated from a z map image or noise, the tint is used as the fog color.

FOG DENSITY

The fog density is used to specify how "thick" the fog is. A value of 1 represents very dense fog. The valid range of values is 0 to 1.

RAMP

The ramp parameter allows you to specify an image file to incorporate into the shader. If a ramp is used, the colour of the fog will by multiplied by the ramp value.

ATTENUATION

The attenuation parameter specifies the distance at which the fog drops its intensity by half relative to the light beam's intensity at its source.

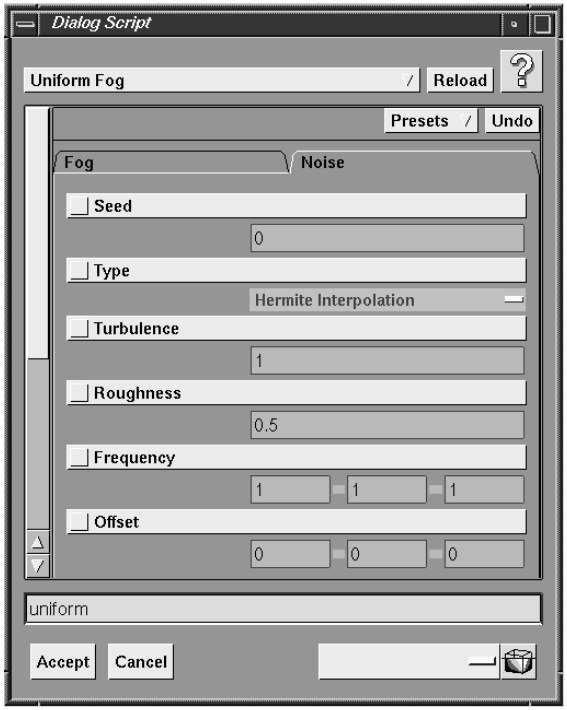
MIN / MAX HEIGHT

For the Layered Fog type, this parameter restricts the scope of the fog on the Y axis.

NEAR / FAR CLIPPING

The near and far clipping planes specify how far away from the light the beam starts and ends.

2.3 PARAMETERS – NOISE PAGE



All atmosphere types share a *Noise* page. The Noise parameters allow you to adjust the character and appearance of the atmosphere in the scene.

SEED

This value is used as a random seed. For each distinct value, a different looking noise field is produced.

TYPE

There are four different noise types: *Hermite*, *Improved Hermite*, *Sparse Convolution*, and *Alligator*. Hermite and Improved Hermite are fastest. The Hermite method uses splines to interpolate values in the noise field. The Improved Hermite method uses a more linear interpolation. These methods contain artifacts at certain positions in the noise field. Sparse Convolution is computationally more expensive, but will produce noise fields free of artifacts present in the Hermite methods. Alligator provides a much different look than the other noise types.

TURBULENCE

The turbulence value represents how many iterations to add fractional distortion to the noise. With each iteration, noise is added that is higher in frequency than the previous iteration. The end effect is that a higher turbulence value will result in more detail in the noise.

ROUGHNESS

The roughness is closely linked with the turbulence. Roughness is used to scale the noise that is added with each iteration of the turbulence. The valid roughness range is 0-1. Lower values will result in smoother noise fields. Values closer to 1 will give coarse noise fields.

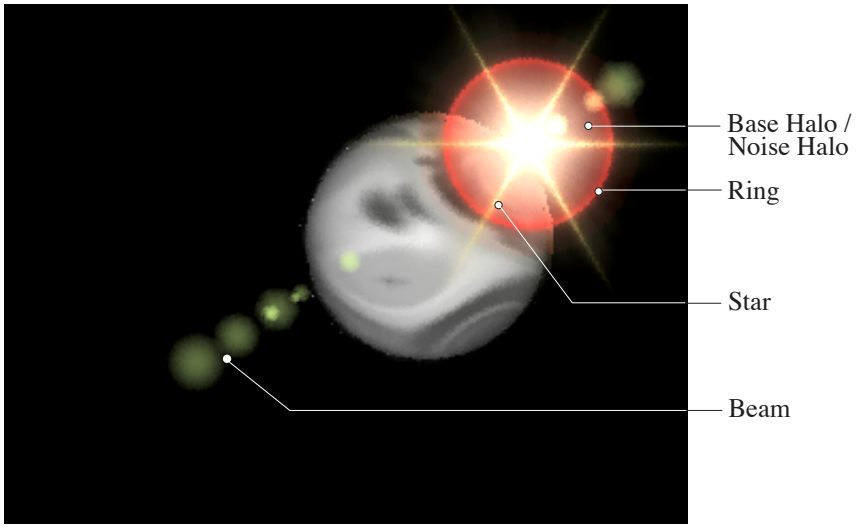
FREQUENCY

Frequency determines the rate at which the noise properties change over a distance in each axis. The frequency essentially scales the noise field. The higher the frequency, the more compressed the noise field will become.

OFFSET

The offset will shift the noise pattern by the offset amount in world space.

3 LENS FLARE SHADER



The values in the lens flare shader are based on a value of one unit sphere located at the origin of the Atmosphere object. Translations made to the Atmosphere object will be passed on to the flare effects created here. For example, rotating the Atmosphere object will also rotate the lens flare.

3.1 PARAMETERS – OCCLUSION SUB-PAGE

NUMBER OF SAMPLES

Determines how extensively the area specified in the *Occlusion Radius* parameter is tested when the flare is blocked by another object. This affects how soft the fading of the flare object is when occluded by another object. There is no computational penalty for high values here, though sixteen produces reasonable results.

OCCLUSION RADIUS

This value represents the portion of the unit sphere sampled for visibility testing in occlusion situations.

3.2 PARAMETERS – BASE HALO SUB-PAGE

BASE COLOR

Determines the colour of the base, or central, halo.

COVERAGE

The coverage parameter determines the opacity of the particular aspect of the lens flare (star, base halo, ring etc.) at the centre of the lens effect. A value of one produces full opacity at the centre, while .5 produces half opacity at the centre of the flare effect.

HEXAGON HALO

A simple toggle, this changes the shape of the base halo.

ROLLOFF

This parameter determines how quickly the halo's opacity goes from one (opaque at the centre) to zero at the edge. A higher value produces a quick transition while a lower value produces more consistent opacity across the halo's radius.

RADIAL RAMP

This parameter allows you to specify a ramp file to produce gradations in colour in the halo. A radial ramp is applied from the centre to the outside of the halo. The colour of the ramp multiplies the base colour.

CIRCULAR RAMP

Circular ramps are applied to the halo beginning along the X axis and proceed around the circumference of the halo.

3.3 PARAMETERS – NOISE HALO SUB-PAGES

The Halo sub-page is identical to the Base Halo page's parameters. The Noise subpage allows for the creation of effects along the halo. If the *Amplitude* parameter is not enabled, the Noise Halo parameters function identically to the Base Halo parameters.

Higher values in the *Frequency* parameter produce flares that are more star-like in shape.

The *Offset* parameter can be animated using expressions.

3.4 PARAMETERS – RING SUB-PAGE

BASE COLOR

Determines the colour of the base, or central, halo.

COVERAGE

The coverage parameter determines the opacity of the particular aspect of the lens flare (star, base halo, ring etc.) at the centre of the lens effect. A value of one produces full opacity at the centre, while .5 produces half opacity at the centre of the flare effect.

HEXAGON RING

This parameter changes the shape of the halo's ring to a hexagon.

RADIUS

The *Radius* parameters determine the size of the ring itself.

RING GLOW

The *Glow* parameter affects the softness of the ring itself.

RING ATTENUATION

Attenuation determines how quickly the glow fades, with lower values producing a softer transition to no glow.

RADIAL RAMP

This parameter allows you to specify a ramp file to produce gradations in colour in the ring. A radial ramp is applied from the centre to the outside of the ring. The colour of the ramp multiplies the base colour.

CIRCULAR RAMP

Circular ramps are applied to the ring beginning along the X axis and proceed around the circumference of the ring.

3.5 PARAMETERS – STAR SUB-PAGE

The star parameters produce a star shape at the centre of the flare. The *Radius* parameter determines the size of the star. Values greater than one extend the star beyond the unit sphere the flare's parameters are based on.

The *Number of Points* parameter will actually produce double the number specified in this field as the calculation is based on a half circle. Two in this field will create a four-pointed star etc. There are no odd angles in the stars created.

3.6 PARAMETERS – BEAM SUB-PAGE

The Beam parameters generate spots that run in a line from the origin of the flare object through the centre of the screen.

3.7 SHAPE SUB-PAGE

NUMBER OF SPHERE POINTS

Determines the number of sphere-shaped points along the beam's path.

NUMBER OF HEX POINTS

Determines the number of hexagonal-shaped points along the beam's path.

OVERSHOOT

The *Overshoot* parameter specifies how long the line of the beam is and affects the distance between hex and sphere points on the beam.

RANDOM SEED

The *Random Seed* parameter determines the distribution of the spheres and hex points on the beam.

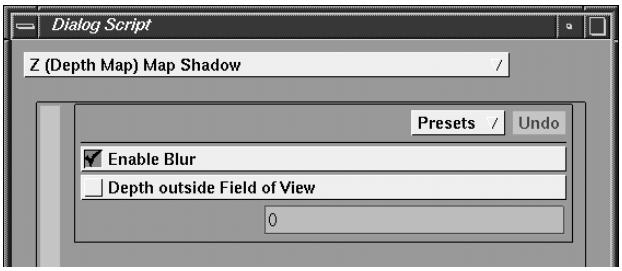
MIN / MAX SPOT SIZE

The two edit fields determine the upper and lower limits of the spots' size on the beam.

3.8 BEAM HALO / STAR SUB-PAGES

The parameters found on the Halo and Star sub-pages are identical to those found on the Star and Base Halo pages. If effect, you are able to control the nature and appearance of the sphere and hex points along the beam with the same precision as the base flare itself.

4 LIGHT SHADER ✣ DIALOG



LOCATION OF DIALOG

Light object > *Shading* page > *Shader* ✣ parameter

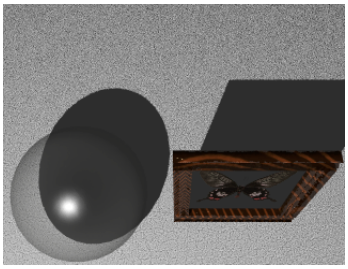
4.1 DESCRIPTION

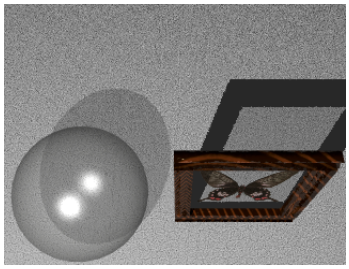
Shadows are a feature of light objects, so setting up and controlling shadows involves modifying the light's attributes. A single *Light* can cast shadows, be a cone light and have a mask (like a slide projector) at the same time. Shadows add realism to your images but they also add to the rendering time, so, to balance the cost and the benefits in your particular situation, the renderers offer many shadow methods.

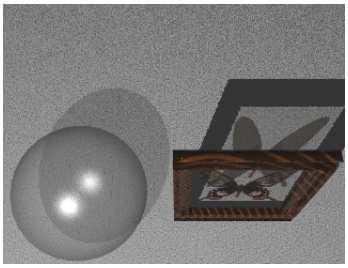
You can decide which lights will cast shadows and which ones won't. For good looking pictures you will usually have only one light which casts shadows and other lights which add highlights and modelling.

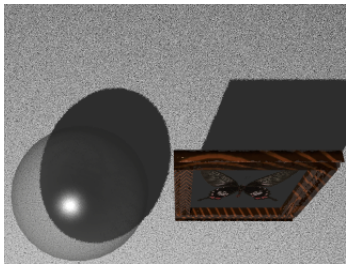
The dialog box associated with this parameter lets you select what kind of shadows are cast during the rendering process.

4.2 LIGHT SHADER TYPES MENU


Fast Shadow


Transparent Shadow


Filter Shadow


Z-Depth Shadow

NO SHADOW

The *noShadow* shader simply produces no shadows at all for that light.

Z-DEPTH SHADOW

The *zShadow* shader uses a depth map to determine visibility. To use this shader, a Z-Depth Map file needs to be specified. The depth map can also be automatically generated at render time with the Auto-Generate Depth Map toggle. Because shadow testing is done by sampling a raster file, this shader is quite fast and also produces blurred shadows.

FAST SHADOWS

The *fastShadow* shader does shadow testing but assumes that all occluding surfaces are opaque, so no transparent shadows will be cast. The shadow edges produced are hard. Because of this assumption, this shader is quite fast.

FILTER SHADOWS

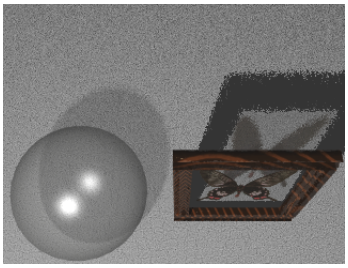
The *filterShadow* shader is similar to *transShadow*. Opacity and transparency colors will be used to affect the shadow. In addition, all texture functions will be used. You must use *filterShadow* to get proper shadows of objects that use the *erode* texture shader.

Note: If you're trying to use *Filter Shadows* to have a texture cast a shadow, you will have to ensure: i) The shader's "Shadow" parameter should be less than one, and ii) In the Texture Map TOP > *Property Mapping* page, you need to set the *Shadow Value* to *Alpha*. With these two set corretly, it will work.

TRANSPARENT SHADOWS

The *transShadow* shader will do shadow testing and, if transparent surfaces are found between the light and the surface, the light will be filtered by the combined shadow-opacity and transparency colors of all surfaces occluding the light. No texture functions are called to calculate any transparency due to maps on occluding surfaces.

4.3 ENABLE BLUR BUTTON



Filter Shadows with *Enable Blur* button enabled

The *Enable Blur* button allows for blurring of shadows cast by objects in the scene. In particular, blurred shadows add realism to the appearance of objects in motion. Truly convincing blurred shadows usually should be accompanied by an increase in the *Shading Quality* parameters in the *Render* tabbed page. This adjustment lengthens the rendering time. The blur value is derived by referencing the Light Object's Shadow Softness parameter. Small values are recommended.

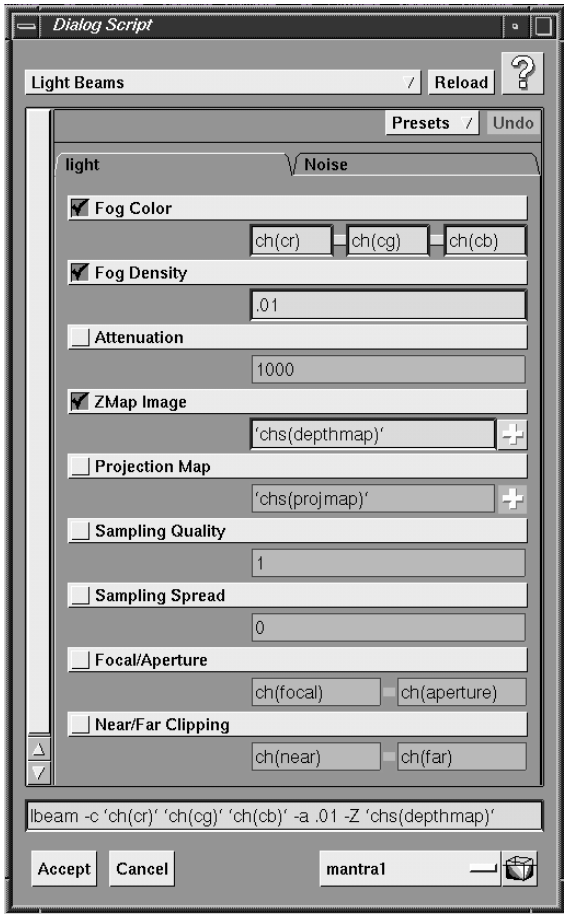
When casting soft shadows, all texture functions are used to filter the shadow. The *Number of Samples* determines how well anti-aliased the shadow will be, and the *Blur Factor* determines how soft the shadow will be.

Note: *Enable Blur* will not work with Micropolygon and Variance anti-aliasing.

DEPTH OUTSIDE FIELD OF VIEW

The *Depth outside Field of View* parameter specifies a value to use when the area to be shaded falls outside the field of view of the light. Setting this to 0 would mean that anything outside the field of view of the light would be in shadow.

5 ATMOSPHERE ✣ DIALOG



LOCATION OF DIALOG

Light Object > *Shading* page > *Atmosphere* ✣ parameter

DIALOG MENU

From this pull-down menu you select the type of light that interacts with the atmosphere.

5.1 PARAMETERS – FOG PAGE

FOG COLOR

Sets a tint color for the light beam. When the light beam is generated from a z map image or noise, the tint is used as the fog color. If a projection map is used, the fog color is the projection map's color multiplied by the tint.

FOG DENSITY

The fog density is used to specify how "thick" the fog is. A value of 1 represents very dense fog. The valid range of values is 0 to 1.

ATTENUATION

The attenuation parameter specifies the distance at which the light beam drops its intensity by half relative to the light beam's intensity at its source.

Z-MAP IMAGE

The Z-depth map image is used to create effects such as light breaking as a result of an intervening object. To get the best results, make sure the focal length/aperture of the light beam is the same as that used to create the zmap image.

PROJECTION MAP

The projection map is an image that is applied across the light beam. It can be thought of like a slide projector. For example. to get a circular beam, apply a projection map that has a circular shape. The alpha of the projection map is used to determine transparency. For best results, the map should have ramped alpha at the edges of the image.

SAMPLING QUALITY

The sampling quality specifies how many times to sample the light beam. Higher values will yield more detail within the light beam but will take longer. If there isn't a lot of variation within the light beam, a small value like 1 will give very good results.

SAMPLING SPREAD

The sampling spread determines the area around the ray that is sampled as it passes through the scene. Increasing the sampling spread generates a softer appearance for the final product.

FOCAL / APERTURE

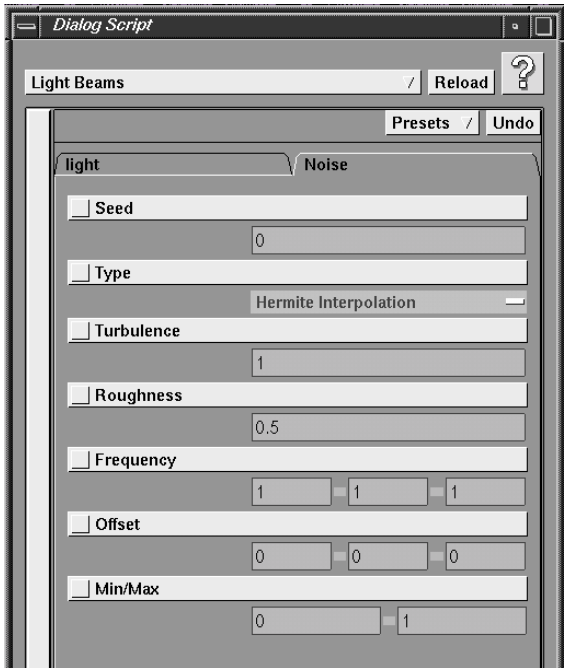
The focal and aperture parameters specify the size of the beam. For best results, these values should be the same as those specified in the light tabbed page. These values (along with the near and far clipping planes) determine light beam's shape.

NEAR / FAR CLIPPING

The near and far clipping planes specify how far away from the light the beam starts and ends.

Note: In order to use the light beam shader, one of the following must be used: a Z-map Image, a Projection Map, or Noise.

5.2 PARAMETERS – NOISE PAGE



All atmosphere types share a *Noise* page. The Noise parameters allow you to adjust the character and appearance of the atmosphere in the scene.

SEED

This value is used as a random seed. For each distinct value, a different looking noise field is produced.

TYPE

There are four different noise types: *Hermite*, *Improved Hermite*, *Sparse Convolution*, and *Alligator*. Hermite and Improved Hermite are fastest. The Hermite method uses splines to interpolate values in the noise field. The Improved Hermite method uses a more linear interpolation. These methods contain artifacts at certain positions in the noise field. Sparse Convolution is computationally more expensive, but will produce noise fields free of artifacts present in the Hermite methods. Alligator provides a much different look than the other noise types.

TURBULENCE

The turbulence value represents how many iterations to add fractional distortion to the noise. With each iteration, noise is added that is higher in frequency than the previous iteration. The end effect is that a higher turbulence value will result in more detail in the noise.

ROUGHNESS

The roughness is closely linked with the turbulence. Roughness is used to scale the noise that is added with each iteration of the turbulence. The valid roughness range is 0-1. Lower values will result in smoother noise fields. Values closer to 1 will give coarse noise fields.

FREQUENCY

Frequency determines the rate at which the noise properties change over a distance in each axis. The frequency essentially scales the noise field. The higher the frequency, the more compressed the noise field will become.

OFFSET

The offset will shift the noise pattern by the offset amount in world space.

MIN / MAX

The min and max values will limit the noise to this range. Both the min and max should be in the 0-1 range. These values can be used to prevent the fog from being completely transparent.

5 The 3D Spatial Audio System

1 3D SPATIAL AUDIO

The 3D Spatial Audio system in Houdini simplifies audio processing by integrating 3D animation and geometry with sound effects. By animating sound sources and microphones within a 3D environment, you can create realistic environmental audio.

1.1 SPATIAL AUDIO COMPONENTS

Environmental audio is broken up into several components in order to easily integrate audio with scene elements.

SPATIAL AUDIO CHOP

This CHOP is the rendering engine for the 3D audio system. It pulls microphones, sound sources and geometry objects together, adds environmental effects and produces the final audio tracks.

MICROPHONE OBJECT

This object acts as a positional listening point. The position, orientation, sensitivity and listening cones are defined within this object.

SOUND OBJECT

This object acts as a sound source. It references the first channel of a CHOP which contains the actual audio to project. Like the Microphone object,it has a position, an orientation, volume controls and projection cones.

ACOUSTIC CHOP

The Acoustic CHOP is used to design filters. It allows you to hand-model spectrum audio filters. It should be used to create all filters used by the Spatial Audio system (at least as a generator).

GEOMETRY OBJECT

Geometry objects can be used as obstacles to occlude sound. They need to have a Sound Material associated with them (which is a set of filters created by the Acous-

tic CHOP). Obstacles should be lightweight in terms of geometry. The Spatial Audio CHOP uses the geometry from the Render SOP.

1.2 SETTING UP A 3D AUDIO ANIMATION

The simplest 3D audio setup is: one Microphone, one Sound source and one Spatial Audio CHOP. The Sound source also requires a CHOP containing the audio to project (often a File CHOP loading an audio file). Any number of microphones and source sources can be used. One audio track is produced for each microphone used, and all the sources are mixed together for each output track.

SCALING THE ACOUSTIC ENVIRONMENT

You may also want to change the scale of the whole environment to reflect the size of the scene. The default scale is 1 meter per 1 Houdini world unit. To make the environment sound larger than the scene, increase the *Meters Per Unit* parameter on the SpatialAudio CHOP's *Environment* page.

CONNECTING COMPONENTS – SPATIAL AUDIO CHOP

The Spatial Audio CHOP is the central hub for all audio components. It:

- References all microphones to be rendered in the *Microphone* parameter. Microphones can reference an optional microphone filter, created with an Acoustic CHOP.
- References all sound sources to be rendered in the *Sound Sources* parameter. Sound Sources reference a CHOP containing the audio to project.
- References any geometry objects to be used in the *Obstacles* paramters. These geometry objects must reference a Sound Material – i.e. a CHOP containing a filter designed by an Acoustic CHOP.

1.3 SIMPLE EFFECTS

These simple effects are quick to render and easy to use. The advanced effects, listed in the next section, take longer to cook and more effort to setup. All of these effects are optional. If all effects are off, the Spatial Audio CHOP behaves like a regular mixer.

DISTANCE DELAY AND THE DOPPLER EFFECT

Sound does not travel instantly, so distant sounds have a noticable delay. This delay also causes the Doppler effect, which is the shifting of pitch as a sound source rapidly passes a listener (e.g. the pitch of an ambulance's siren lowers as it passes).

The amount of delay is determined by the speed that sound travels. Usually you will be using the speed of 331 m/s, which is the speed of sound in air (at zero degrees Celcius, but room temperature is very close to this also). Another useful speed to know is 1452 m/s – the speed of sound in water (at 15 degrees Celcius). The speed

of sound is defined in the SpatialAudio CHOP's 'Speed of Sound' parameter in the Environment page.

If you want to exaggerate the delay or the Doppler effect, lower the speed of sound from its realistic value. A speed of 166m/s will double the delay and the Doppler shift in air.

VOLUME LOSS OVER DISTANCE

Sound also loses intensity as it travels. The SpatialAudio CHOP supports two methods for determining the volume loss at any distance.

using realistic dropoff

This method is the easiest to set up, and produces a realistic result (it is based on a phyics equation, *power = 1/[distance^2]*). To change how fast this dropoff occurs, you can modify the '10m Distance Loss' parameter on the SpatialAudio CHOP's Effects page. This parameter specifies the amount of sound intensity loss after travelling 10 meters. A value of 0.4 means that the sound will loss 40% of its intensity after 10 meters (because of the dropoff curve, after 20 meters the sound loss will be at 85%).

using a dropoff table

This method requires you to create a curve that represents the dropoff of the sound as it travels farther. This curve can be created with another CHOP and passed to SpatialAudio CHOP by connecting it to SpatialAudio CHOPs first input.

The *Volume Lookup Range* parameter determines the range of distances that this curve describes. If this parameter is set to (5, 100), then the first value of the curve will be the volume level at 5 meters, and the last value of the curve will be the volume at 100 meters. Anything outside the lookup range will be sampled in the curve's extend regions.

This method is a little more work to setup, but allows you to create any kind of dropoff, even completely unrealistic ones.

ECHOES

Echoes and environmental reverb can be produced with very little effort. The SpatialAudio CHOP supports two types of echoes; static environment echoes and dynamic echoes. Static environment echoes are a good choice for enclosed areas, while dynamic echoes simulate open areas well.

Both echo methods allow you to specify the number of echoes, the echo delay (or the minimum echo delay), and the echo volume.

The echo dropoff is determined by the same dropoff method as normal sounds. If the sound delay is 0.1 seconds, and the speed of sound is 331m/s, the echo travels 33 meters farther than the original sound, so the volume drops according. Each echo has the same delay from the previous echo (so the 4th echo arrives 0.4 seconds after the original sound if the echo delay is 0.1 seconds).

If you find the echoes are too subtle, increase the *Echo Volume* parameter.

using static environment echoes

Static echoes always have the same delay between echoes, regardless of the distance between the source and the listener.

using dynamic echoes

Dynamic echoes base their echo delay on how far the original sound travelled. Distant sounds will have longer pauses between the echoes and close sounds will have almost no echo delay.

The 'Dynamic Echo Effect' parameter determines the portion of the travelling delay that is used as the echo delay. So, if the Dynamic Echo Effect is set to 0.3, and the sound travelled 4 seconds to reach the listener, the echo delay will be 1.2 seconds (0.3x4).

To ensure a minimum echo delay, set the 'Echo Delay' parameter to a non-zero value. This will add a static echo delay to the dynamic echo delay. So an Echo Delay of 0.1, Dynamic Echo Effect of 0.3 and a travelling time of 4 seconds would produce an echo delay of 1.3 seconds (0.3x4 + 0.1).

1.4 ADVANCED EFFECTS

These effects require filters (see the next section, *Designing an Audio Filter* p. 373 for more details). Most of the simple effects are done by the Spatial Audio CHOP; these effects require other objects and CHOPs, so they take slightly longer to set up — but they're worth it.

OBSTACLE OCCLUSION

This effect allows you to create geometry object and use them to block sound. "Blocked" sound can be completely blocked, partially blocked or even amplified. A filter is used to describe which frequencies are blocked by the object, and by how much.

setting up an object for 3d audio

Any geometry can be used as an obstacle. Most obstacles should be fairly low resolution geometry, since increasing the resolution has little effect on the overall audible result.

Each obstacle needs a Sound Material associated with it, to describe how it affects the sound passing through it. If the Sound Material includes a transmission filter, the filter is used to filter the sound directly. Otherwise, the absorption or reflection filter is used (but inverted to determine the transmission filter). See the next section, Designing a Filter, for details on filters.

To add the geometry as an obstacle to the SpatialAudio CHOP, enable Check For Obstacles on the SpatialAudio CHOP's Effects page. Then add the geometry object to the parameter 'Obstacles' (using the pull-down menu). Objects without Sound Materials will be ignored.

tweaking the occlusion effect

The parameter 'Obstacle Softness' determines how soft the obstacle is. Soft obstacles allow quite a lot of sound around them, while hard obstacles do not (softness of zero). Two of the main factors that affect how much sound get by the obstacle are:

- how far both the sound source and the microphone are from the obstacle (more sound gets around the obstacle at longer distances)^
- the size of the obstacle (larger objects allow less sound around them)

So, an object between a sound source and the microphone that are close together will block most of the sound, while the same object roughly in the middle of a widely separated source source and microphone will have little effect. A larger object will always have more of an effect than a smaller one in the same situation.

If you don't want any sound to get around the obstacles, set the obstacle softness to zero.

MICROPHONE FILTERS

Each microphone listens to the full spectrum by default. However, you can setup a microphone to only listen to a certain parts of the spectrum. This allows you to create microphones which create subwoofer tracks, high frequency tracks, or midrange tracks. This effect is created by adding a filter to a Microphone Object.

The Microphone object has a parameter 'Filter' on its Microphone page which references a filter. The filter should be created by an Acoustic CHOP, and it should be an Absorption filter ("absorb"). See *Designing an Audio Filter* p. 373 for more details.

THE ECHO FILTER

The echo filter is an optional filter that is applied to each echo. This allows you to change the sound characteristics of the echo.

The SpatialAudio CHOP must be generating echoes for this filter to have any effect. To add an echo filter, connect the CHOP containing the filter to the third input of the SpatialAudio CHOP (the 'Echo Filter' input).

The filter is applied more than once to higher numbered echoes. The first echo has the filter applied to it once, the second has the filter applied to it twice, the third three times, and so on. Because of this, you should not make the filter block sound too drasticly (with values close to 0).

See *Designing an Audio Filter* p. 373 for more details.

THE ENVIRONMENT FILTER

The environment filter is an optional filter applied to all sounds. The effect of the filter varies with distance, exactly like 'Volume Loss over Distance'. It uses the same method to calculate the volume loss. The filter is multiplied by the calculated volume loss and applied to the sound.

When designing an environment filter, you can use either a transmission or absorption filter to describe the volume loss. You should not make the losses too large (not more than 0.5), otherwise sounds will lose most of their intensity over a very short distance.

To add an environment filter, connect the CHOP containing the filter to the SpatialAudio CHOP's second input (the "Environment Filter" input). See the next section, *Designing an Audio Filter* p. 373 for more details.

2 DESIGNING AN AUDIO FILTER

Many of the SpatialAudio effects require a filter. This section describes how to build filters with the Acoustic CHOP, and what different types of filters there are.

2.1 WHAT IS A FILTER?

You can think of a filter as a series of volume controls per frequency. There are several types of filters. The term 'Pass' means that a frequency is left mostly untouched as it passes through the filter, and other frequencies are suppressed (they are 'stopped').

ALL PASS AND ALL STOP FILTERS

These filters are the simplest. An *All Pass* filter passes all frequencies equally, and acts like a normal volume control. An *All Stop* filter does not pass any frequencies; it is simply an *All Pass* filter with a value of zero.

LOW PASS FILTERS

These filters pass low frequencies, but not higher frequencies. They usually have a transistion period where increasing frequencies are partially passed. These filters are used to remove noise and mute sound. They normally make audio sound muffled. Low pass filters are excellent for modelling obstacles (like walls).

HIGH PASS FILTERS

High pass filters are exactly the opposite of low pass filters. They pass high frequencies but not low frequencies. They tend to make audio sound tinny. High pass filters are an excellent choice for modelling atmospheric sound loss and reflections.

BAND PASS FILTERS

Band pass filters pass mid-range frequencies, and not high or low frequencies. They are sometimes referred to as notch filters.

BAND STOP FILTERS

Band stop filters suppress mid-range frequencies and pass both high and low frequencies. Both Band Pass and Band Stop filters aren't quite as useful for SpatialAudio as the previous filters, but they can be used for special cases.

2.2 USING THE ACOUSTIC CHOP TO DESIGN A FILTER

The Acoustic CHOP is a modelling tool for creating audio filters. It allows you to create a full or partial spectrum filters by interactively pulling handles. Several filters can be created at once, with the added benefit that multiple filters can be constrained to preserve total signal power. Filters designed with the acoustic CHOP can then be manipulated by any other CHOP to help model the filter.

TRANSMIT, ABSORB, REFLECT

The first three parameters, Transmit, Absorb and Reflect Sound, toggle the transmission, absorption and reflection filters on or off. Normally, you will want to create one filter of a specific type.

FREQUENCY RANGE

The next parameter, Frequency Range, determines the range that the filter will cover. The beginning of the channel corresponds to the start of the range, and the end of the channel corresponds to the end of the range.

POWER AND LOCK

The Power and Lock parameters are for conserving power when creating several different filters for the same object. If an object reflects some sound and transmits the rest, you may only want to design the reflection filter and have the transmission filter automatically adjusted so that the total signal power is conserved.

power

The Power parameter can be turned off for unconstrained modelling, set to "Conserve Total Power" so that the total signal power is conserved (but not the individual frequency power), or "Conserve Power Per Frequency" which ensures that each frequency has its own power conserved. The Lock parameter allows you to lock one of the filters so that it is not affected by the automatic power conservation algorithm (mainly for use with 3 filters). If you increase one filter beyond 1, then power will no longer be conserved.

NO RESONANCE

The No Resonance parameter constrains the filter so that filter values cannot be increased beyond 1. This prevents the filter from adding power to the signal.

HANDLES PAGE

The Handles page contains all the parameters for modelling. The Handles parameter specifies the number of interactive handles that are created. This can be changed on the fly, but may result in some data loss.

Since audio filters are logarithmic in nature, it is easier to shape filters with logarithmically spaced handles. These will give you more control in the lower frequencies, and less control on higher frequencies. They are enabled by default, but you can use evenly spaced handles by toggling Logarithmic Handles off.

SAMPLES

The Samples parameter determines how many samples are in the actual filter. You can increase the filter accuracy by increasing the number of samples if you notice your filter is lacking resolution in areas of high detail.

INTERPOLATION

The Interpolation parameter specifies how the samples of the filter are interpolated from the handles (none, linear or cubic). Cubic produces smooth filters and None produces notched filters.

You can reset all the waveforms to constant values by clicking Reset All Waveforms. If you want to clear a single filter, but not the rest, toggle that filter off, then on using the toggles on the Acoustic page.

6 Character Tools

This section discusses the tools that have been specifically designed to aid in the creation and animation of characters. These tools are referred to, collectively, as the *Character Tools* and consist of:

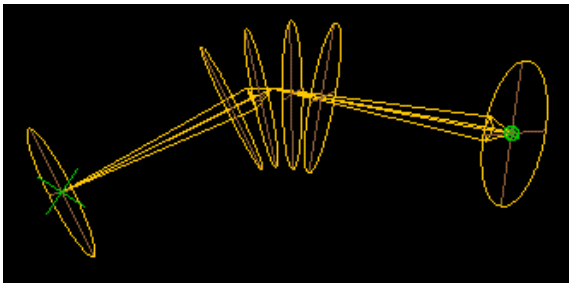
- Bone objects
- InverseKin CHOP
- Bones Operation
- Capture Geometry Operation
- Edit Capture Regions Operation
- Edit Capture Weights Operation
- Mirror Operation
- Mirror Capture Weights Operation
- Paint Capture Weights Operation
- Pose Operation

1 BONE OBJECTS

The Bone Object is the foundation of all of the Character Tools. It is an Object with most of the properties of a Geometry Object. It also has some extra features such as length, two types of geometry, end-to-end linking, and kinematic parameters.

1.1 BONE GEOMETRY

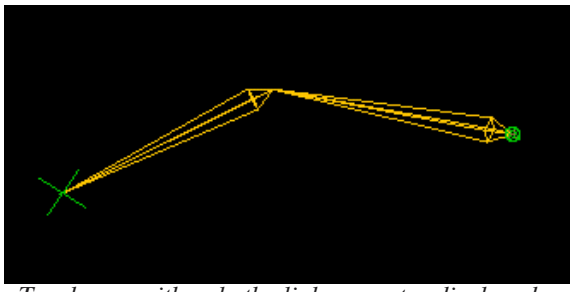
By default, Bone Objects contain two types of geometry: *link geometry* and *capture geometry*. Both of these are defined in the Bone Object's SOP network and are therefore customisable. In most cases, the defaults are sufficient.



Two bones with both types of geometry displayed

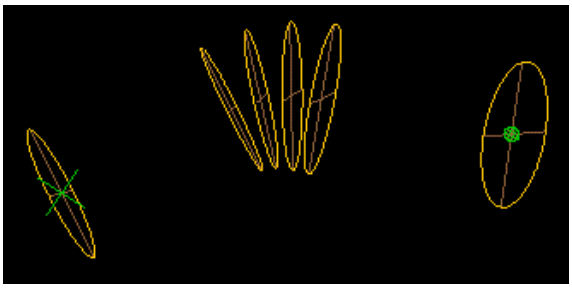
The link geometry consists of a narrow diamond shape with one corner at the origin of the Bone Object. The opposite corner is stretched down the Bone Object's negative z-axis in accordance with the value in the Bone Object's length parameter. This

link geometry is used to simply give a visual indication of the position and size of the Bone Object.



Two bones with only the link geometry displayed

The capture geometry defines a capture region that can be used to apply skeletal deformation to some other geometry by using a Skeleton SOP (refer to *Skeleton OP* P. 745 in the *SOPs* section for details on skeletal deformation). It consists of two ellipses that are positioned along the length of the Bone Object. There are several parameters in the Bone Object that can be used to adjust the size and position of each of these ellipses. If a Bone Object is a direct descendant of another Bone Object, then its capture geometry will also contain a user-defined number of ellipses connecting the first ellipse of the child to the second ellipse of the parent. These ellipses provide extra control over skeletal deformation of the "joints" of a character.

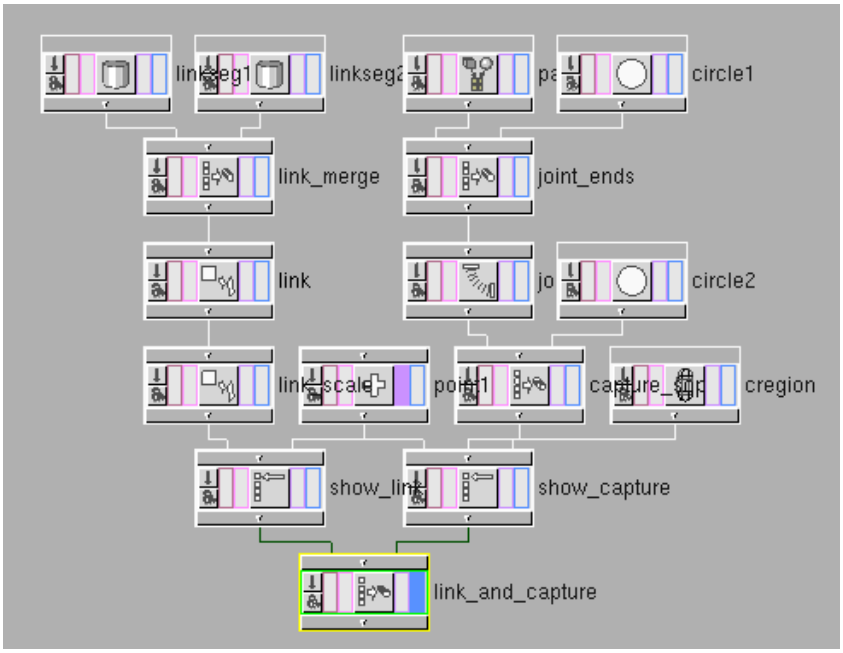


Two bones with only the capture geometry displayed

The link geometry and capture geometry have separate parameters controlling whether they are displayed. Usually, the link geometry is displayed while doing the positioning of the bones, and the capture regions are displayed when defining the capture regions used for skeletal deformation. When the character has been set up fully, then both display parameters can be turned off.

BONE OBJECT SOP NETWORK

The construction of both types of geometry in the Bone Object's SOP network is fairly straightforward and worth investigating.



The link geometry consists of two pyramid shaped meshes positioned base to base, and stretched with a Transform SOP which references the Bone Object's length parameter. The result is then fed into a Switch SOP that will switch between the link geometry and nothing, depending on the value of the Bone Object's *Display Link* parameter.

The capture geometry is created using two Circle SOPs that reference the appropriate parameters in the Bone Object to determine their size and orientation. Their size depends on both the *Ellipse 1 & 2 Radius* and *length* parameters of the bone object. This is so that the default capture regions will be reasonably sized for commonly used bones. The *length* dependence can be easily removed by modifying the expressions in each circle's *radius* parameter.

The joint of the capture geometry is created using a Joint SOP which was created for this specific task. It creates intermediate ellipses between the last ellipse of the parent Bone Object (which is imported using the Object merge SOP called *parent_end*), and the first ellipse in the current Bone Object (circle1).

Note! In order for the Joint SOP to function correctly, the parameters of *parent_end* must be set manually by the user, unless the bone-creation Operation was used to create the bone.

The output of the Joint SOP is combined with *circle2* to define the capture geometry of the Bone Object. This geometry is fed into another Switch SOP which controls the display of the geometry in a fashion similar to the Switch SOP used for the link geometry.

There is an Add SOP called *point* which outputs a single point. This SOP is used as the replacement for the link and the capture geometries when they are not being displayed. It is also used as the Bone Object's render geometry, as its render flag is set. In other words, it is not because of this SOP that Bone objects don't render. If some part of the bone geometry needs to be rendered, the render flag of the SOP that contains the geometry should be set instead.

1.2 END TO END POSITIONING

By default, when a bone is parented as the child of another bone object, it will automatically use the tip of the parent bone as its origin. For all other objects, the origin will be the same.

1.3 BONE PARAMETERS

Bone parameters are discussed in *Bone Object* p. 267. However, for detailed discussion of the parameters pertaining to kinematics, refer to *Character Tools* p. 376.

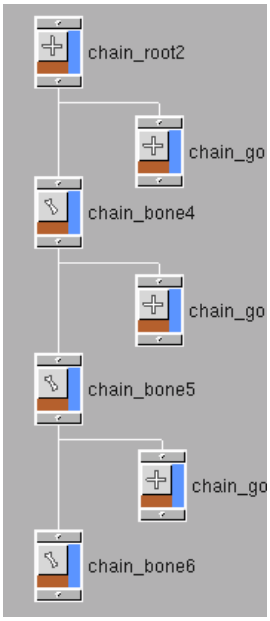
It should be noted that some of the common transform parameters (scale, pivot, transform order and rotate order) do not appear in the Bone object's parameter list, however, they do exist and can be changed using the channel editor or keyboard commands. Bone positioning and kinematics will only function correctly if these values are unchanged. Here are the parameters and their correct values:

Scale	1, 1, 1
Pivot	0, 0, 0
Transform Order	Scale Rot Trans
Rotate Order	Rz Ry Rx

1.4 CREATION OF BONES

While bones can be created one by one, we highly recommended that the Bones Operation be used. The Bones Operation greatly simplifies the process of creating bones. It sets up Null objects at the base of bones when necessary so that the kinematic solutions will behave better.

2 BONE CHAINS



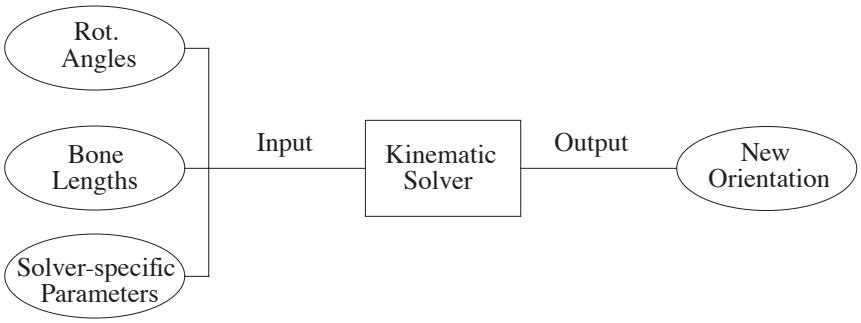
WHAT IS A BONE CHAIN?

Any linear hierarchy of Bone Objects can be organized into a Bone Chain. Once a Bone Chain is defined, it is associated with a Kinematic Solver (the IK CHOP) which determines the orientation of each Bone Object in the Bone Chain.

For example, you could use a *Follow Curve* Kinematic Solver which aligns the bones in the Bone Chain along a given curve. This and other Kinematic solvers are discussed in *Kinematic Solvers* p. 381 below.

THE IK CHOP

The inputs to all Kinematic Solvers are the length of each Bone Object in the Bone Chain, and its orientation when "at rest". From its inputs, the IK CHOP computes the appropriate orientation of the bone chain, and returns that as output.



The rest angles of a chain are generally used to specify the initial orientation of the chain from which the solution is derived. The rest angle for each bone in the Bone Chain is contained in the *Bone* page in that bone's parameters. The extra solver-specific parameter is specified in the same *Bone* page, but only in the Driver of the Bone Chain.

Whenever the IK CHOP cooks, it collects all the rest angles from each of the bones in the Bone Chain, and passes them, along with any solver-specific parameters, to the appropriate Kinematic Solver. It then takes the output from the solver and sets the rotation parameters of each bone in the bone chain appropriately.

Note that because the Driver is setting each bone's rotation parameter, the rotation parameters shouldn't be animated by the user. They will always be over-written by the values given by the IK CHOP.

3 KINEMATIC SOLVERS

3.1 INTRODUCTION

There are four types of kinematic solvers in Houdini. Kinematic solvers are responsible for determining the orientation of bone chains. Each will orient a given bone chain according to the algorithm it uses and its given parameters.

- Inverse Kinematics
- Follow Curve Kinematics
- Show Rest Position
- Show Capture Position

The first two solvers are used to control bone chains for animations, while the last two are used as "helper" utility tools.

Each of these solvers will be discussed in this section. In addition, a method of bone chain control called *Forward Kinematics* will be covered.

3.2 INVERSE KINEMATICS

In Inverse Kinematics, we specify the desired location of the end of a bone chain and the solver will subsequently determine the corresponding set of joint angles within the bone chain. The desired location of the bone chain is specified using the origin of an object. The IK solver tries to bend the bone chain so that its end touches the origin of the object. This object is referred to as the end-affector of the chain. The start of the bone chain (the location of the start of the first bone) is called the base of the chain.

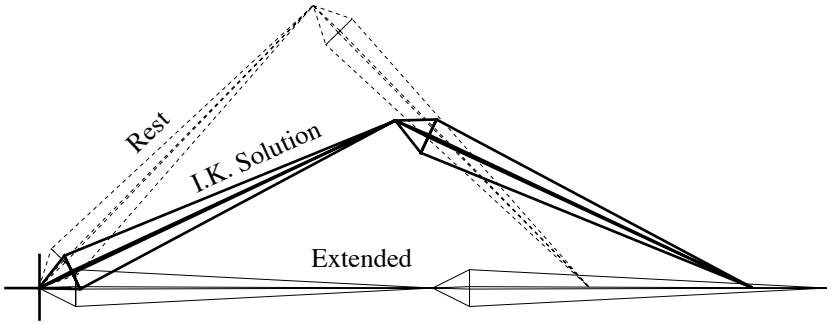
Houdini's IK solver produces unique solutions. That is, for a given end-affector position, the same solution is always found regardless of the direction of movement of the end affector. Secondly, the shape of the IK solution can be changed interactively by changing the rest angles of the bone chain. Thirdly, Houdini's IK is prismatic, which means that you can animate the bone lengths within a bone chain and the IK solver will continue to solve for the end-affector position. Finally, Houdini's IK can work with planar IK chains (2D) and non-planar IK chains (3D).

HOW DOES HOUDINI DETERMINE INVERSE KINEMATICS SOLUTIONS?

Consider the following situation: An IK chain is fully extended (all of its bones are in a straight line) and the end affector is then moved closer the to base of the chain. The joints in the chain must bend to allow the end-affector to move in. The question is, "How does Houdini decide which way to rotate the joints in this situation?" The answer is that you specify the preferred direction of rotation by using the *Rest Angles* of the bones.

REST ANGLES

Simply put, each joint in an IK chain will rotate toward the rest angle. Thus, in the figure below, if a chain has the rest angles shown in dashed lines *(Rest)*, the chain will move toward the configuration shown in bold lines *(I.K. Solution)* when compressed from full extension.



Rest angles are always 0 by default, since they are taken implicitly from the *pretransform*. If rest angles are non-zero, then they are an additional rotation on top of the pretransform. A far easier way to adjust rest angles is to adjust the position of the joints at the capture frame using the Bones operation.

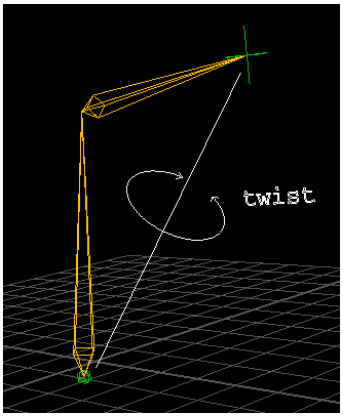
3.3 ADDITIONAL CONTROLS FOR IK

THE BONE LENGTH

This parameter controls the length of the bone segment and can be animated. As the bone length changes, the IK solver will adjust the internal joint angles so that the end of the chain remains stuck to the end-affector. Note however that if you have channels set for the bone length and you change the bone length, the IK solver will not cook until a keyframe is set (the red keyframe indicator is changed from red to green).

THE TWIST

This parameter specifies a rotation about the axis which runs from the base of the bone chain to the end-affector. It is very useful for controlling the twisting of the ball joint in at the top of the chain.



3.4 FOLLOW CURVE KINEMATICS

Houdini's Follow Curve kinematics provides a method of controlling bone chains with many segments, such as in spines or tails.

Follow Curve kinematics requires a piece of curve geometry to follow (the Curve Object parameter). This curve can be of any type (NURBS, Bézier or Polygon). Follow Curve will adjust the joint angles within a chain so that the bone chain follows the shape of the curve. If the curve is too far from the bone (that is, the bone cannot

physically reach the curve), the joint angles of that bone are adjusted so that the distance between the end of the bone and the curve is minimised.

A follow-curve chain with curve away from the first two chain bones

Tip: When you have a follow curve kinematics bone chain, you can animate the twist of the bones via the *Rest-Z* parameter on the bones.

3.5 SHOW REST POSITION

This kinematic solver will look up the bone angles in the "Rest Position" parameter of each bone and use these angles to rotate each bone. This solver is useful if you wish to view the orientation of your bones when they are in the rest position.

3.6 SHOW CAPTURE POSITION

This kinematic solver is similar to *Show Rest Position* except that it uses the angles in the "Capture Angles" parameter to determine the orientation of each bone.

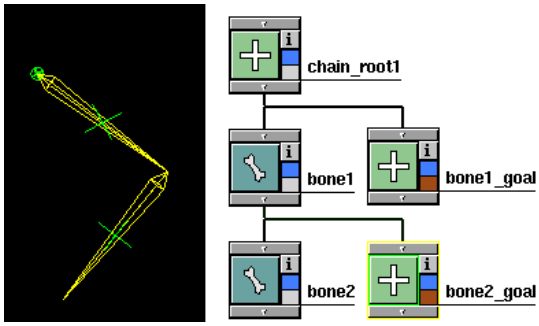
The capture angles of the bone object store the orientation of the bone when the bone was last used for capturing points in the Skeleton SOP. The Show Capture Position solver can be used to position the bone chain as it was at the time of the capture so that capture regions can be adjusted.

3.7 FORWARD KINEMATICS CONTROL

In Forward Kinematics, we specify the rotations of each bone and thus we inherently move the end of the skeleton to some place in space. In this sense, Forward Kinematics is not really a kinematic solver.

There are two ways you can control forward kinematics in Houdini. The first is to create a bone chain hierarchy and simply change the angles in the Transform Tab of each bone object. Keep in mind that bone objects are still Houdini objects and can thus be parented and transformed in the same way that geometry, cameras and lights are.

A second method of control for Forward kinematics is creating a hierarchy of single-bone inverse kinematic chains. The end-affectors for each single-bone IK chain can then be used to directly manipulate the joint rotations in the view-port.



A forward kinematic chain with affectors and hierarchy.

This type of Forward kinematics using affectors is set up automatically when you create a Forward kinematic chain using the Bones Operation.

4 BONES OPERATION

4.1 DESCRIPTION

The Bones Operation is one of Houdini's Character Tools. It enables you to interactively create bone chains by drawing in a Viewport. Some special features of the Bones Operation include:

- The creation of capture regions, bone objects, and end affectors automatically
- Easy parenting of a new bone chain to other geometry
- Control over the naming of new geometry
- The replacement of standard bone geometry with user geometry
- Explicit numeric specification of bone positions
- Automatic colouring of bone chains.
- Easy positioning of the joints of bones.

For a complete description of the Bones Operation, see *Bones Operation* p. 310 in the *Interface* section.

5 DEFORMING GEOMETRY WITH BONES

5.1 INTRODUCTION

Houdini has three SOPs which define the way geometry is deformed by bones. This addresses the problem of assigning point deformation weights (termed "weighting" by some other packages). This provides the most procedural, intuitive and flexible way to weight a character and deform it.

5.2 OVERVIEW

CAPTURE REGION SOP

The Capture Region SOP defines a volume (a tube with a hemisphere at each end). Points which are inside this Capture Region will be deformed as the Capture Region moves.

CAPTURE PROXIMITY SOP

The CaptureProximity SOP uses distance instead for capturing. It weights each point based on the closest bones.

DEFORM SOP

The Deform SOP actually deforms you geometry (as the name implies) based on the point capture attributes in the geometry. The weighting and deformation of geometry were separated into two SOPs to give you the flexibility of procedurally modifying the geometry and/or weighting between these two operations.

Please see the *Geometry* section for complete information on these SOPs.

7 RBD Tools

1 INTRODUCTION TO RBD

Houdini includes the ability to perform rigid body simulations. This functionality is provided through a combination of POPs, CHOPs, and new Object parameters. There is also a new RBD Operation in the Object Editor to allow quick and easy setup of rigid body simulations.

1.1 SOME DEFINITIONS

Following, are some terms that are used in the rest of this chapter, but which may require some explanation:

RBD

RBD stands for: Rigid Body Dynamics.

RIGID BODY

A rigid body is a mathematical representation of a real world object. Rigid bodies differ from real world objects in one extremely important respect. Rigid bodies do not bend, break, squish, compress, expand, or flow in response to forces or collisions or other contact with other rigid bodies. Their shape is in no way affected by what happens to the object. In Houdini it is possible to specify a deforming object as a rigid body, but that deformation is defined with no respect for what is happening in the rigid body simulation.

RBD SOLVER

The RBD Solver is the piece of code in Houdini that handles rigid body simulations. The RBD Solver is built into POPs.

RIGID BODY PARTICLES

A rigid body particle is a regular Houdini POP particle that has a special flag checked in the pstate attribute of the particle. Such particles are controlled by the RBD solver, completely separately from non rigid body particles. Most RBD particles also have a wide assortment of other attributes, some of which are used by non-RBD particles (such as angular velocity, torque, and mass) and some of which are only used by RBD particles (dynamic and static friction).

ACTIVE / PASSIVE

All rigid bodies can be divided into two classes: active and passive. Passive rigid bodies are not affected by collisions with other rigid bodies, regardless of their mass. Examples of passive rigid bodies would be a floor or a train. You can also use passive rigid bodies to keyframe the path of certain objects either for part of a simulation of for an entire simulation. However, passive rigid bodies need not have large masses or be stationary. You can specify a rigid body as passive but with a small mass so that external forces and torques will still affect its motion. Active rigid bodies are affected by collisions with other rigid bodies.

I.2 THE RIGIDBODY POP

The key to creating rigid body simulations in Houdini is the RigidBody POP. This POP lets you create rigid body particles, and set relevant properties for these particles. It can also take input from another POP to convert existing particles into rigid body particles. A rigid body particle differs from other particles in several ways.

- A rigid body particle has a number of attributes which other particles may or may not have. These attributes are use by the rigid body solver (such as torque, static and dynamic friction, center of mass, instance).

- Rigid body particles can perform particle-particle collision detection using the instance attribute of the particle to determine the geometry to use for the collision detection. This collision detection changes the angular and linear velocities of the colliding particles to simulate collisions between ideal rigid bodies, taking into account bounce, relative masses and inertial tensors to create realistic motions.

- Rigid body particles are controlled by a different solver than other particles. The rigid body solver does not allow the suppression of rules (using the Suppress POP). This is because to generate realistic motion and avoid object interpenetration, the solver must be allowed to control all aspects of the particle's motion.

  Rigid body particles, however, do react to subsequent POPs the same way as other particles. If you append a Force POP to a network that contains both rigid body and normal particles, both types of particles will respond to the force in the same way (assuming similar masses on the particles).

BIRTH PARAMETERS

generation objects

This parameter contains a list of objects. For each object in the list, a single particle is birthed at the origin of the object. The particle instance attribute is set to the object that generated it.

birth time

Time at which to birth particles for the Generation Objects.

RIGID BODY PARAMETERS

exclusive groups

The rigid body solver provides a very flexible mechanism for specifying which rigid body particles should be tested for collision. Simply specify the names of existing point groups in this field. All pairs of rigid body particles that share membership in one or more of these point groups are tested for collision. Group names entered in this field are global and apply to all rigid body particles in the POP network, not just the particles generated by this RigidBody POP. If no exclusive groups are set in the network, all rigid body particles do collision detection with all other rigid body particles.

make input particles rigid bodies

If this check box is off, input particles to this POP are not affected in any way. This ability is useful if you simply want to specify collision groups without actually changing any particles. The parameters of the first page (Generation Objects) are still created if this parameter is off.

source group

If "Make Input Particles Rigid Bodies" is on, this parameter specifies the group of particles to operate on.

particle geo

If "Make Input Particles Rigid Bodies" is on, this parameter specifies the value that should be assigned to the instance attribute of input particles, and thus which object should be used to inherit attributes for the particles.

attributes page parameters

The Attributes page lets you specify which attributes should be inherited from the particle instance object. The initial position and velocity attributes are only inherited on the first frame of the simulation. All other inherited attributes are updated from the instance geometry at each frame. Thus parameters that are animated at the object level will be animated at the particle level as well. See the section on New Object Parameters below for detailed descriptions of all these new parameters.

1.3 PASSIVE RIGID BODIES

If a rigid body is active, or it is passive and has less than infinite mass (an object is considered to have infinite mass if its mass is greater than 1e+38, or its mass is exactly 0.0), its motion is controlled by the RBD solver. The RigidBody POP creates the particle and sets the initial velocity, angular velocity, rotation and position, and then doesn't affect the particle after that. If however the rigid body is passive (the Object's Active parameter is 0, or the particle Active state was set to 0 with the State POP) and has a mass greater than 1e+38 (or a mass equal to 0.0), the RigidBody POP treats it differently. The RBD Solver in this case does not move or change the velocity of the particle. The RigidBody POP looks at the keyframed positions and rotations of the object and sets the particle attributes appropriately at each frame. Thus you can have rigid body particles that are actually keyframe ani-

mated, but will cause reactions in other rigid body particles. Pairs of keyframed rigid body particles are not tested for intersection because neither object would be affected by the collision.

1.4 THE CONSTRAINT POP

The Constraint POP lets you create geometric relationships between rigid body particles that are maintained by applying forces and torques to those objects. Constraints are not guaranteed to be maintained, especially when one of the objects in the constraint is involved in a collision. This is because collisions cause instantaneous changes in velocity, where constraints can only apply forces. Thus the object may be involved in a collision which changes its velocity, but the constraint doesn't have a chance to apply a corrective force until the next frame, at which point the particle has already moved, possibly violating the constraint. However constraints are very good when there are multiple, possibly conflicting, constraints applied to a single object. For example, if a single object has two constraints, one which tells the object to go to (0,0,0), and the other which tells it to go to (1,0,0), the object will quickly come to rest at (0.5, 0, 0).

Each constraint requires you to specify one or two particles or groups of particles on which the constraints should operate. The Group fields let you specify point groups (or list point numbers) that the constraint should be applied to. The Object parameters let you specify the name of an object. The instance attribute of each particle is compared against this value and the constraint is applied to the first particle that matches this object. If more than one particle is specified, a constraint is generated for each particle. The constraint parameters can use local variables to specify different values for each particle.

There are three constraint types: point to nail, point to point, and rotation. All parameters to all types of constraints can be animated. Each constraint type is described in more detail below.

POINT TO NAIL

A Point to Nail constraint holds an object in a fixed position in space. There are two parameters required to specify this constraint. The Nail Point is the point in world space that the object is constrained to. The Object Point specifies the point on the object (relative to the particle position) that should be constrained to the specified point in space. So if the Nail Point is specified as (1,1,1), and the Object Point is specified as (0,0,0), forces will be applied to the particle to move it to (1,1,1). Notice that this constraint says nothing about the orientation of the object. If the Nail Point is specified as (1,1,1) and the Object Point is specified as (1,0,0), forces will be applied to the particle so that the particle is one unit distance from (1,1,1), and the object will be oriented so that a vector of (1,0,0) drawn from the particle position would end at (1,1,1). This type of constraint can be visualized as a solid rod connecting a point in space to a point on the object.

POINT TO POINT

A Point to Point constraint holds one object in a fixed position relative to another object. This type of constraint requires you to specify a point relative to the position of the first object, and a point relative to the position of the second object (Object A Point and Object B Point). This constraint is similar to the Point to Nail constraint except instead of a nail, the object is fixed to another object. This constraint can be visualized as a rod connecting two objects.

ROTATION

A Rotation constraint holds an object in a particular orientation. This constraint is specified with a vector in world space (the Rotation Vector) and a vector specified relative to the object (the Object Vector). Torque is applied to the object so that these two vectors remain parallel. This constraint does not in any way affect the position of the particle, only the orientation. Also, the object will still be able to spin freely around the axis specified by the Object Vector.

COMBINING CONSTRAINTS

You can use combinations of the above constraints to simulate more complex real world relationships between objects. For example, a hinge can be created using a rotation constraint and a point to nail constraint.

1.5 THE PARTICLE CHOP

The Particle CHOP generates translation and rotation channels to move an Object according to the position and rotation of the rigid body particle that instances that object. This CHOP is the glue that ties a rigid body simulation back into objects. It can be used in two modes: Cached or Dynamic. In Dynamic mode, the POP network is cooked only when the current frame is advanced, and it is cooked only up to the current frame. In this mode simulations cannot be played in reverse, and cannot be edited. The simulation is re-initialized whenever the play bar is reset to frame 1. In Cached mode, the entire frame range is cooked and cached. In this mode the simulation is only cooked when the Update button in the Particle CHOP parameter dialog is pressed. Thus you can safely make any number of changes to your network and not worry about re-cooking with each change. Only when you are done all of your modifications do you hit the Update button to refresh the cache. This CHOP also contains all the standard parameters used when cooking a POP network. See the section on POP Cooking Parameters for details.

1.6 THE UNIFYRBD POP

The UnifyRBD POP lets you create a single RBD particle from a number of existing RBD particles. This makes it very easy to produce effects such as a single object breaking into pieces, or multiple objects sticking together. The key parameter in this POP is the Source Group. All particles in the Source Group must be RBD particles or an error is reported.

The UnifyRBD POP creates a single particle. The mass, position, velocity, forces, and torques on this particle are calculated based on the masses, positions, etc. of the input particles. The RBD solver then solves for the motion of the unified particle and propagates these new attributes back into the sub-particles. There may appear to be some inconsistencies in the attributes of unified RBD particles (for example, the rotation attribute of unified RBD particles is always 0). This is because the set of sub-particles that make up a unified particle can change completely from frame to frame, so the unified particle must be essentially destroyed and recreated at each frame. This is acceptable because it is only the motion of the sub-particles that is really important to the animation. Only the sub-particles represent real objects in your scene. The unified particles are just a handy way of grouping these objects together and breaking them apart.

A unified RBD particle is very different from a group of objects with constraints between the objects. The relative positions of sub-particles in a unified RBD particle are absolutely fixed (constraints have some flexibility, especially during collisions). With a unified particle, no collision detection is done between sub-particles of the unified particle. Unified particles are also much faster than constraints because no special force calculation is required to maintain the relative particle positions. This is just a consequence of the way that unified particles and sub-particles are treated by the RBD solver.

The UnifyRBD takes a reference input, so it is possible to create hierarchies of RBD particles. For example, say you had 9 RBD particles, one for each of obj1-obj9. You could create a particle unify1 for obj1-obj3, unify2 for obj4-obj6, and unify3 for obj7-obj9. Then you could create a grandUnify particle from unify1-unify3. Also very useful is the fact that collision information is maintained at all levels. In the situation above if obj1, obj3, and obj7 have collisions on a given frame, collision attributes are set (if requested in the RigidBody and UnifyRBD POPs) for obj1, obj3, obj7, unify1, unify3, and grandUnify. The collision information for unified particles is calculated as the sum of the collision information of all its sub-particles. And because you can request collision information on a per particle basis, you can have only the unified particles contain the collision information, or only the sub-particles, or both, or neither.

Constraints that are applied to sub-particles are lost when the sub-particle becomes part of a unified particle. You must constrain the unified particle itself if you wish to apply constraints to unified particles.

You can choose to have the unified particle belong to all the groups that any sub-particle belongs to. This is in case you are using the "Exclusive Groups" feature of the RBD solver. If this option is chosen, the unified particle will be tested for collision with all particles that any of the sub-particles would have been tested against.

1.7 OBJECT PARAMETERS

Geometry objects have a new page of parameters that can be accessed by the POP network to set attributes for rigid body particles. The parameters (located on the Physical page) are to be fonud in *Parameters – Physical Page* p. 290.

1.8 THE FORCE OBJECT

The Force Object is a place holder for parameters that are used by a number of POPs. It allows you to use direct manipulation in the viewport to control, from the object editor, some of the forces in a POP network.

1.9 THE RBD OPERATION

The RBD Operation automatically builds POP and CHOP networks to set up and run rigid body simulations. It creates a POP network that contains a RigidBody POP, and a CHOP network that contains a Particle CHOP. It can also add other POPs to apply forces or constraints, and can modify the parameters of these OPs to add or remove objects to the simulation, and control the forces on the objects.

1.10 POPS TO KNOW ABOUT

STATE POP

The State POP includes several new parameters specifically for use with RBD particles. The Rigid Body parameter lets you force certain particles to be RBD particles, or to stop being RBD particles. If you set this value to 1 for some particles, the RBD solver becomes responsible for controlling the motion of those particles. If the particles do not have all the required attributes or contain bad attribute data (such as an empty instance attribute), the particle may not move or behave as you would expect. The Active parameter lets you change RBD particles from active to passive rigid bodies. The Override parameter lets you enable or disable the creation of channels for the particle's instance object if the POP network is passed through a Particle CHOP.

PROPERTY POP

The Property POP has also been extended to include new parameters specific to RBD particles. These parameters are Center of Mass, Bounce, Dynamic Friction, and Static Friction. These parameters affect the same attributes that are controlled by the object parameters with the same names.

FORCE-APPLYING POPS

Several POPs now contain a field to specify a Force Object. In all cases, if a force object is specified, the parameters of the POP become disabled and the POP parameters are gathered from the Force Object. This feature is used by the RBD Operation to allow control of POPs through the viewport. But you can also use this new mechanism yourself to allow control of POP networks (even non-RBD networks) by manipulating 3D geometry in the viewport.

GROUP POP

The Group POP supports a new grouping mechanism based on instances. You can use this mechanism to group particles quickly and easily based on their instance attribute. This serves mostly as a convenience when creating RBD networks because it is easier to identify RBD particles based on their instance attribute than by point number or using an expression to check instance values.

1.11 LOCAL VARIABLES

All of the POP attributes relating to RBD have local variables to access their values. In the following table are also listed some attributes and local variables that already exist, but are specifically useful when using RBD.

RBODY	pstate - flag that the particle is an RBD particle.
ACTIVE	pstate - flag that the particle is an active RBD particle or a passive RBD particle.
OVERRIDE	pstate - flag that the particle instance geometry transform should be overridden by a Particle CHOP.
MASS	mass - Mass.
ROT	rot - Rotation.
W	w - Angular Velocity.
TORQUE	torque - Torque.
COM	com - Centre of Mass.
BOUNCE	bounce - Bounce.
FDYNAMIC	fdynamic - Dynamic Friction.
FSTATIC	fstatic - Static Friction.

1.12 POP COOKING PARAMETERS

The RBD solver requires a number of new parameters to specify how to cook the network. These parameters can be found on the Rigid Body Dynamics page of the Particle CHOP, the POP SOP, and the POP Viewer parameter dialog. The following sections describe each of the new parameters.

REST THRESHOLD

This parameter is used to specify the relative velocity between objects that is considered as resting contact by the RBD solver. Resting contact is treated differently from collision contacts, which is why this division is necessary. Usually the default value should be used here unless your objects are very large.

MAX TIME SPLITS

This parameter specifies how finely the RBD solver is allowed to split up a time step. The solver splits time steps to find the precise moment at which two object collide. The larger the number of time splits you allow the more accurately collision can be simulated, but also the longer the simulation may take. This parameter should take into account the maximum velocity at which particles will be moving in the simulation, the Contact Tolerance value, and the size of each time step (which is one frame time divided by the Oversampling parameter on the Standard page).

CONTACT TOLERANCE

This is the distance between two objects at which the RBD solver considers them to be touching. If this value is made large, objects will collide and bounce in obviously unrealistic ways (i.e. before they actually contact). If this value is made small, you should increase the Max Time Splits value which will allow more accurate simulation of collisions but will also slow the simulation.

CONSTRAINT TIME

This parameter specifies the approximate time (in seconds) that the RBD solver can take to satisfy an unsatisfied constraint (see the Constraint POP for information about setting up constraints). The smaller this value is, the more rigidly constraints behave, but the larger the forces are that get applied to the objects. The larger this value is, the less likely you are to produce an unstable situation in your simulation.

SOLVER TYPE

There are three supported solvers: Euler Solver, Midpoint Solver, and Runge-Kutta Solver. They are listed in order of decreasing speed and increasing stability. This parameter is rarely important unless you are using constraints, and should be kept on Euler unless you find your constraints producing unstable behavior in your simulation.