imagine
Imagine...
vision
model
Try this....
Imagine
3D
...CUP addendum
render

This is a contract. If you proceed to download and/or install the files in this software, you accept all the terms, conditions and obligations imposed upon you by this contract.

This Software product is a copyright (c) 1999 of Impulse, Incorporated. All title and copyright in and to the Software (including but not limited to any images, animations, and text incorporated into the Software) are owned by Impulse, Incorporated.

You are granted a limited license to the use of a single copy of the Software for non-commercial purposes on only a single personal computer or workstation which is not used as a server, provided that you agree to the following:

You may not make any copies of the Software or any part of it.

The Software is owned by Impulse, and its structure, organisation and code are the valuable trade secrets of Impulse, Inc. You agree not to modify, adapt, translate, reverse engineer, decompile, disassemble or otherwise attempt to discover the source code of the Software, or to give, rent, lend, sell or lease the Software to anyone.

The Software is made available for your use “as is” and without any warranties, including any warranty of merchantability, any warranty of fitness for a particular purpose, any warranty as to non-infringement of the rights of third parties, or any other warranty whether expressed or implied.

You assume the entire risk of using the Software. Any liability of Impulse, Incorporated will be limited exclusively to product replacement or refund of purchase price. In no event will Impulse, Incorporated be liable to you or third parties for consequential, incidental or special damages, even if Impulse, Incorporated has been advised of the possibility of such damages.

This Agreement will be governed by the laws of the State of Nevada, excluding the application of its conflicts of law rules and the laws of the United States. This Agreement will not be governed by the United Nations Convention on Contracts for the International Sale of Goods, the application of which is expressly excluded. If any part of this Agreement is found void or unenforceable, it will not affect the validity of the balance of the Agreement.

You acknowledge that you have read this Agreement, understand it and that by downloading or using the Software, you agree to be bound by all of the terms and conditions of this Agreement.

If you have any questions about this Agreement, please contact Impulse, Incorporated in writing or by fax.

Impulse, Inc.

7250 Peak Drive Suite 102

Las Vegas

NV 89128

Telephone: **702 948 1100**

Facsimile: **702 948 1104**

Email: **sales@coolfun.com**

Forward

This addendum to the I Files details the additions to ***Imagine*** for Windows over the course of the "Constant Upgrade Programme" — the intermediate versions between IFW 1 and IFW 2. Subscribing to the "Constant Upgrade Programme" allows users to see and be involved in the evolution of ***Imagine*** as new features are added, many of which have been specifically requested by users. There is of course, a substantial discount to users who participate in the programme.

The addendum is broken down into two main parts: an overview and a detailed look at the enhancements made to ***Imagine*** for Windows. For those enhancements made over the course of the CUP which have already been covered in the I Files, references have been included to the appropriate sections.

Thanks must go to Zack Knutson (chief programmer of ***Imagine***) for taking the time to explain how and why he has implemented certain features. For those familiar with the "Hitchhikers Guide to the Galaxy", "Deep Thought" spoke of something far more intelligent to come. I think he was referring to Zack, whose merest operation parameters, I am not worthy of comprehending…

I can only hope I have done justice to what he said.

Richard Foster

April 1999

One point that should be noted is that many of the enhancements made over the course of the "Constant Upgrade programme" pave the way for a radical overall in the interaction between ***Imagine*** and the user. As such, certain options may not have "evolved" fully. Participation in the next "Constant Upgrade Programme" will allow you to ensure they mature the way you want them to!

Contents

Legalities..........2

Forward..........3

Contents..........4

Overview

CUP… features..........6

OpenGL, enhanced Color display mode, interactive Lighting, Gamma correction, Point/edge/face count limit removed, Alpha channel, Transparency mapping, Render Frame, enhanced Set Zone, Anaglyph, stage States, enhanced Bones, Fog falloff, "Real life falloff" lights, Overdrive attributes, animatable Deform Tool, Import/Export filters

CUP… commands..........11

REDUCE POINTS, REVERSE PICK, REVERSE HIDE, DRAG ZOOM, MIRROR, SNAP TO GRID

CUP… interface..........13

Primitives drag and drop toolbar, Primitives portal, Attributes portal, F/X portals, Scrub-bar, Font preview, Action Dialog markers, Moveable sub-rows, Numeric readouts, Keyboard shortcuts, View identification axes

CUP… miscellaneous..........17

additional Primitives, Photoshop Plugin stage F/X, Jiggle stage F/X, enhanced, Ripple stage F/X, enhanced Pick Subgroup, revised Textures

Details

OpenGL.....19
interactive Lighting.....25
Alpha channel.....29
Transparency mapping.....37
Render Frame.....41
Set Zone.....48
Anaglyph.....49
refined Bones and Kinematics.....71
Import/Export filters.....75
Reduce Points.....78
Primitives drag and drop toolbar.....84
Primitives portal.....86
additional Primitives.....87
Attributes portal.....94
F/X portals.....99
Jiggle stage F/X.....104
Enhanced pick subgroup.....108

Overview

CUP… features

OpenGL

OpenGL..addendum reference: 19

Support for the OpenGL graphics display library has been added for the *Perspective view* and the new *Portals*. Where a suitable graphics adaptor providing hardware acceleration is not available, the OpenGL display will be rendered under software interpretation.

enhanced Color display mode

24Bit/16M Colors...addendum reference: 24

Following the introduction of OpenGL, it was found that where this was being run under software interpretation the display may not be responsive enough. Consequently, the original proprietary high speed 216 Color display mode has been enhanced to full 24 bit support. This may provide a more responsive display where hardware acceleration for OpenGL is not available.

interactive Lighting

interactive Lighting...addendum reference: 25

Coinciding with the introduction of the OpenGL display mode (but not limited to it) an interactive Lighting system has been introduced for both the *Perspective view* and the new *Portals*. Being able to interactively move the light source (rather than having to rotate the object) may assist the modelling process by highlighting areas of interest.

Gamma correction

Gamma tab..I Files reference: 9~8

The inclusion of Gamma correction has improved the appearance of ***Imagine***'s renders significantly. Graphical controls are available for fine-tuning this feature to suit individual monitors.

Point, edge and face count limits removed

Limits tab..I Files reference: 9~7

Whilst there is still a theoretical maximum, limits on the geometric complexity of any single object have been removed. This major change means that objects saved in the new "32 bit" format are no longer compatible with older versions of ***Imagine***. Additionally, a revised set of Stage F/X modules have been supplied to cope with the new object format.

Alpha Channel

Alpha Channel..addendum reference: 29

The addition of Alpha Channel support allows the generation of an associated transparency mask for seamless image composition or video overlays. The implementation of this feature also allows brushmaps to be selectively applied using their own Alpha Channel; controlled from the new *Transparency tab*.

Render frame

Render Frame..addendum reference: 41

Alternatively known as "render area" or "safe area", this command gives an accurate visual indication of the area in the scene that will be rendered by overlaying a marquee on the display in the *Perspective view*. Controls are provided to tailor the marquee to the aspect ratio of the final render.

enhanced Set Zone

Set Zone..addendum reference: 48

Following on from the introduction of the Render Frame, the older SET ZONE command is now visually indicated, also by a marquee overlaying the display in the *Perspective view*.

Anaglyph

Anaglyph tab...Addendum reference: 49

The new Anaglyph 3D stereo viewing feature has been added allowing both still pictures and animations to be rendered as colour, greyscale or pure Anaglyphs. Viewed with a pair of red and blue/green lens glasses, Anaglyphs convey a true sense of volume in a scene.

stage States

Edit Group..I Files reference: 7~64

This new command allows access to "child" objects in a grouping hierarchy from within the *Stage editor*, thus allowing the use of forward and inverse kinematics to pose characters in their staged surroundings.

Group State Bar...I Files reference: 7~75

State sub-row timeline..I Files reference: 7~101

The GROUP STATE BAR command stores the "pose" of the currently picked grouping hierarchy in the State sub-row timeline. This information is stored with the Staging file and thus allows the other State Data Types to co-exist with the new, but over-riding stage Grouping state data type. When using *Detail editor* States, the State sub-row determines which State the object assumes in each frame as well as velocity and spline interpolation information.

refined Bones and Kinematics

refined Bones and Kinematics..............................Addendum reference: 71

Dealing with “Boned” objects is now much easier with the constraints pop-up dialog, axis identification markers and the saving of contraints within the object file.

Fog falloff

global Fog falloff...I Files reference: 7~127

Fog falloff when applied to the global Fog parameter allows the density of fog to with height, rather than the previous uniform density.

object Fog falloff...I Files reference: 5~27

Falloff when used with Fog objects allows the density of fog to vary radially, along a specific axis or along a specific plane formed by two axes. A Hot Center option also allows the intensity of the fog to be significantly increased towards the centre of the object.

New falloff type for light sources

1/RR (Real Life) object lights.....................................I Files reference: 5~40

1/RR (Real Life) stage lights.......................................I Files reference: 7~37

With the new Gamma correction feature, more reliance should be placed on the “falloff” light sources. To this end a new lighting type has been added, which accurately simulates point or spherical artificial light sources.

Overdrive attributes

Specular overdrive...I Files reference: 5~10

Fog length overdrive...I Files reference: 5~31

Clipping tab...I Files reference: 9~12

The Overdrive function allows the intensity of specular highlights and/or the Fog colour to be boosted, by acting as a multiplier on the base value. This however, can cause an odd optical illusion where the base value is not the same in all three channels; hence the inclusion of the *Clipping tab* to minimise any distortion.

animatable Deform Tool

animatable Deform Tool..I Files reference: 2~132

The Deform Tool is a spline "control cage" that can be used to deform an object to which it is applied. Previously, the Deform Tool could only be used to "statically" deform an object's geometry according to how the spline "cage" was manipulated. However, it can now be used to "dynamically" deform any object that comes under its influence in the *Stage editor*. Additionally, since the Deform Tool is, in many ways treated like any other object, different arrangements of the spline "control cage" can be stored as States, allowing the deformations to be animated.

Import/export filters

Import/export filters...Addendum reference: 75

Whilst the existing DXF import/export filter does allow objects to be exchanged with other programs, it can be a cumbersome and unwieldy format. Consequently, two new import and export filters have been introduced for two of the more popular "other" 3D modelling and animations programs, namely 3D Studio and Lightwave.

CUP... commands

Reduce points

Reduce Points...addendum reference: 78

The REDUCE POINTS command displays one of the new *Portals* allowing the interactive optimisation of overall object geometry. This new command will also operate selectively, on a picked selection in either the Pick POINTS, Pick EDGES or Pick FACES editing modes. Real-time feedback (depending on the complexity of the object) allows the results of the optimisation to be viewed as the point count is reduced.

Reverse pick

Reverse Pick..I Files reference: 2~39

Available in all editing modes, this new command picks all of the currently displayed items not picked, and un-picks all of those picked. The REVERSE PICK command is available to all editing modes.

Reverse hide

Reverse Hide...I Files reference: 2~40

Whilst HIDE POINTS editing mode is active (see page 2~44 in the I Files) this command hides all of the currently displayed items and displays all of those currently hidden.

Drag zoom

Drag Zoom...I Files reference: 1~20

This new command is in effect an interactive version of the ZOOM RATIO command (see page 1~18 in the I Files) allowing a precise area of the scene to be magnified to fill the active *View*. Depressing the *Shift* key whilst using the mouse will zoom out.

Mirror tool

Mirror Tool..I Files reference: 2~68

In the past, to “flip” (mirror) an object about one of it’s axes — or the world axes — negative scale values had to be supplied in the *Transformation dialog*. The MIRROR TOOL simplifies this process by allowing the same thing, but from the dedicated *Mirror dialog*.

Snap to grid

Snap to Grid..I Files reference: 1~23

The SNAP TO GRID command differs from the continuously operating GRID SNAP (see page 1~23 in the I Files) in that it is a one-off command that forces the selected item to jump to the nearest grid intersection.

CUP interface

Primitives drag and drop toolbar

Primitives drag and drop toolbar..........................Addendum reference: 84

This new floating toolbar streamlines the workflow inasmuch that all the pre-defined objects are now graphically displayed in a single toolbar. The chosen object can then be "dragged and dropped" onto the *Editor*.

Primitive portal

Primitive portal..Addendum reference: 86

Once a primitive has been chosen from the drag and drop toolbar the *Primitive portal* is displayed. Not only are the parameters for determining the object's geometric complexity available, but the surface attributes can also be specified at this stage. The important advantage to using a *Portal* is that it gives instant visual feedback on how the chosen parameters affect the object's size, complexity, geometry and edge treatment.

Attributes portal

Attributes portal...Addendum reference: 94

Selecting the ATTRIBUTES command now displays the *Attributes portal*. The advantages of this new *Portal* over the older *Dialog* is the improvement in the display of the surface attributes as they are applied to the object, stemming from the "mini" perspective view in the *Portal*.

F/X portals

F/X portals..Addendum reference: 99

Applying the Portal treatment to all the Stage F/X now makes it possible to see what effect the sometimes "obscure" parameters have on the object or object group. The real-time visual feedback provided by the *Portal* will, of course, depend on the complexity of the object and the speed of the computer.

Scrub-bar

Scrub-bar..I Files reference: 7~20

The inclusion of this new *Stage editor* toolbar is really to improve the process of animation, grouping as it does, all the most commonly used animations commands. It also houses what is sometimes known as a "scrub" bar allowing you to quickly navigate to any frame defined within the current project.

Font preview

Font preview..

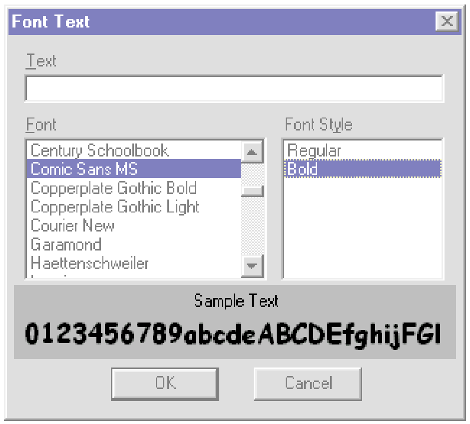


Figure 1: The enhanced Font Text dialog showing the sample text for the selected font and style.

The Font preview is a small but significant enhancement making the selection of the desired typeface far easier. Using the ADD FONT STRING command in the *Spline editor* displays the *Font Text dialog* (see page 4~10 in the I Files) in which the required text is supplied and the font and style is chosen. Previously, the selection of the font relied on users knowing which one was required. Now, with a preview being shown in the Sample Text area of the *Dialog*, it is just a case of browsing through the fonts available to the operating system.

Action dialog markers

Action Dialog markers..

A simple interface enhancement that highlights the frame number and the current sub-row being referenced. This allows the various time-line bars to be positioned more accurately and quicker.

Moving sub-row bars

Move..I Files reference: 7~91

Sub-row bars can now be moved anywhere within the current timeline by selecting the ***Ctrl*** key and dragging the bar. Again, an effective enhancement that simplifies the use of what was a more complicated operation.

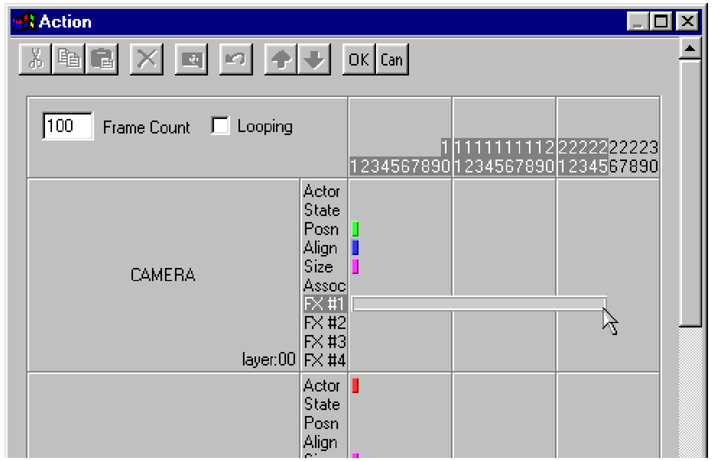


Figure 2: The Action Dialog showing the markers highlighting the sub-row and frame numbers being referenced by the cursor.

Numeric readouts

Numeric readouts...

Whilst sliders are easy to use due to their graphical nature, a problem arises when trying to place them at the same level in subsequent work sessions. The inclusion of a numeric readout alongside the slider solves this, providing a reference as to the exact location of the slider. The anti-aliasing Threshold Value and Ray Limit sliders in both the *Quickrender Settings* and *Render Setup dialogs* now have associated numeric readouts, as has the Shutter Time slider in the *Render Setup dialog*.

Keyboard shortcuts

Keyboard shortcuts

I Files reference: throughout

All the most often used commands now have keyboard shortcuts associated to them, as shown alongside the menu entry. Using a keyboard shortcut to execute

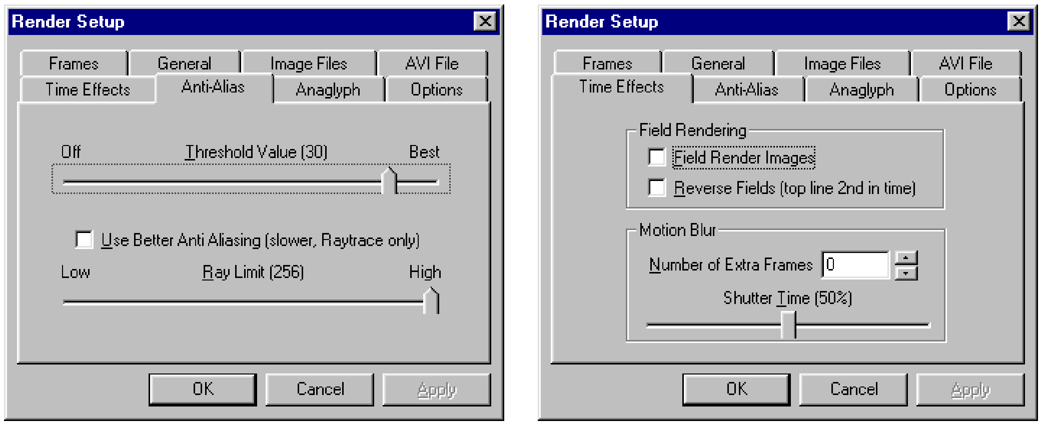


Figure 3: The sliders on the Anti-alias and Time Effects tab showing their associated numeric readouts.

a command can considerably improve the work rate when compared to wading through the various menus and sub-menus.

View identification axes

View identification axes..I Files reference: 1~12

The three *Fixed views* in each *Editor* now have identification axes in the corner to remind you of that *View*'s orientation to the world co-ordinate system.

CUP miscellaneous

additional Primitives

Hemi primitive..addendum reference: 88

Box primitive..addendum reference: 90

Rod primitive..addendum reference: 92

To coincide with the introduction of the *Primitives drag and drop toolbar*, these three new primitives have been added. Whilst each could be created previously, it involved using several commands; each of the new primitives therefore streamline the modelling process. Additionally, appropriate primitives now benefit from a Sharp Edges option, which again, saves a good deal of time.

Photoshop plugin stage F/X

Psplugin F/X..I Files reference: 10~60

This powerful new Global stage F/X allows access to many of the hundreds of Adobe Photoshop compatible post-processing effects. This allows many such effects to be applied at the time of rendering, thus providing better integration and saving time.

Jiggle F/X

Jiggle F/X..Addendum reference: 104

Another new stage F/X was developed during the course of the "Constant Upgrade Programme". Jiggle applies itself to single objects or object groups allowing such effects as wobbling jelly, swaying curtains and undulating water beds to be simulated.

enhanced Ripple stage F/X

Ripple F/X..I Files reference: 10~32

The Ripple stage F/X now benefits from a new parameter called Continuous Waves. This produces either continuous radial or linear waves for the entire duration of the F/X.

enhanced Pick Subgroup mode

Pick Subgroup..Addendum reference: 108

The *Pick Subgroup dialog* has now been upgraded to a "tree" based *Dialog* in which all the subgroups defined for the currently picked object are listed. Operating in either the Pick POINTS, EDGES or FACES editing modes, either the whole subgroup or just the boundary or interior elements that comprise it can be picked. Perhaps more significantly, as each subgroup or part thereof is highlighted in the *Dialog*, the points, edges or faces that are being referenced are highlighted in the *Editor*, thus providing a visual clue as to what is be selected.

revised Textures

revised Textures...

Most, if not all of the textures provided with ***Imagine*** have had their functionality improved, allowing a greater range of values to be supplied and parameters to be deactivated.

Included along the way were many refinements to existing functions, most notably Filter and of course many bug fixes. Impulse would like to thank all those members of the ***Imagine*** for Windows "Constant Upgrade Programme" who took the time to pin-point those undocumented features otherwise known as bugs.

With the release of ***Imagine*** for Windows the highest quality of display in the *Perspective view* was the proprietary flat shaded representation of the objects or scene in 216 colours, derived from it's DOS and Amiga predecessors. During the course of the Constant Upgrade Programme the display in the *Perspective view* — and indeed, the Render window in the newly introduced "*Portals*" — has been dramatically improved with the adoption of the industry standard…

OpenGL

OpenGL is the computer graphics industry's most widely used and supported application programming interface, or "API". Essentially, all this means is a common set of routines (Graphics Library) are used to render (display) the object being described.

Introduced in 1992 and developed from Silicon Graphics Inc. IRIS GL, OpenGL was designed from the start to provide maximum access to suitably enabled display adapters for hardware acceleration. In addition, OpenGL also allows a software driver to compensate for any or all of the functions not catered for in hardware.

In essence, OpenGL is a robust set of routines that interpret an object's geometry — as provided by the application — and renders it for display; it describes how an object should be rendered as opposed to describing the object itself. This interpretation can be software based relying on the host computer's processing power, hardware based relying on specialised functions in the display adapter or a combination of the two. Obviously, "real-time" display (depending on the complexity of the objects) is more likely to be achieved with a suitable display adapter. However, the quality of display using software interpretation is a significant step forward.

To provide as near "real-time" display as possible, the routines that comprise the OpenGL standard are relatively "low-level". This does restrict what OpenGL rendering is capable of. Specifically, rendering is said to be "local" in that when describing how a particular object should be rendered, OpenGL has no knowledge of that object's environment. Thus, reflective and refractive surfaces (realised by scanline and raytrace rendering) are not accommodated. The same is true of most of the available surface attribute parameters, indeed, only the Base Color, Specular Color and whether or not Phong Shaded has been selected for the object are covered by OpenGL's rendering. Nonetheless, the rendering provided by OpenGL is fast and accurate, leading to a quality display, using as it does Gouraud shading (adopting the colours assigned at the points of the polygon to smoothly interpolate across it).

It is probably easiest to think of OpenGL as 3D graphics equivalent of Adobe's "Post-script". Both are platform independent, both can take advantage of dedicated hardware to rasterise the display (be it for the screen or a printer) and both interpret the host applications description of the object/page to render an image.

As with many "so called" standards, they evolve over time; OpenGL is a prime example. Version 1.2 is the current implementation, approved by the OpenGL Architecture Review

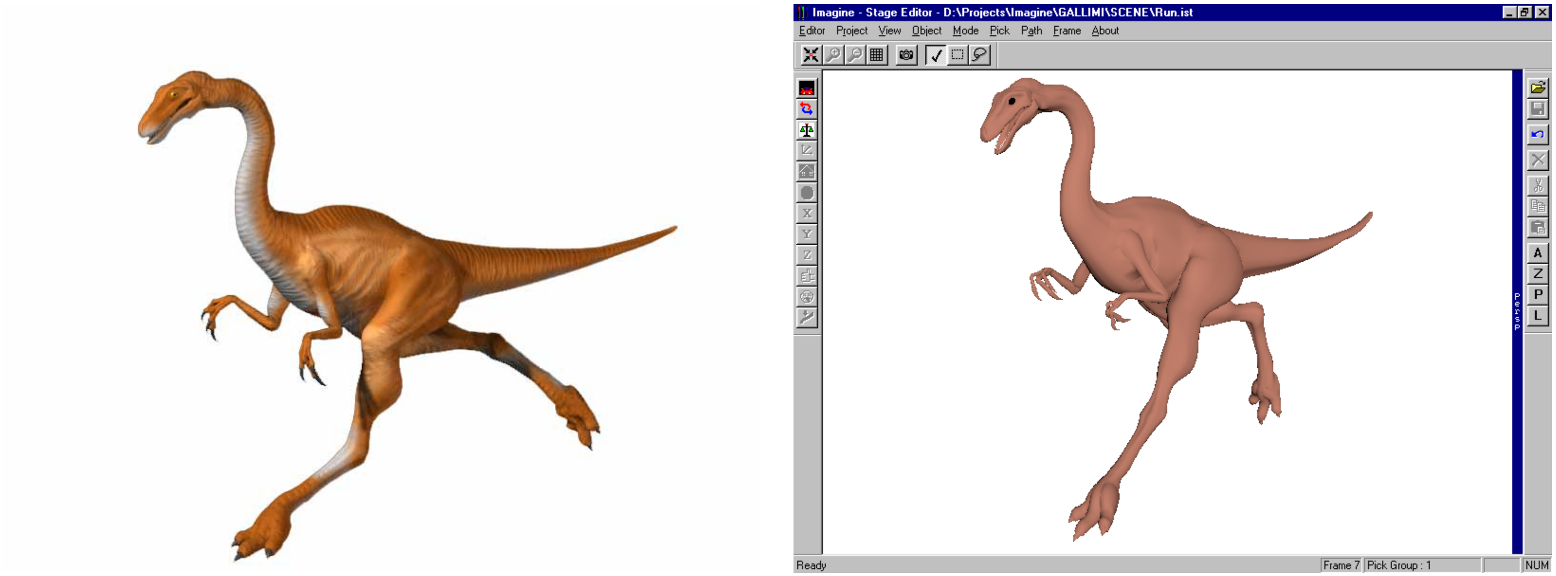


Figure 4: Views of the Gallimimus object from Mauro Marenzi's Dinosaurs CD-ROM. On the left is a raytraced rendering and on the right is a screen shot of the Perspective view using OpenGL with the Color option enabled.

Board. This independent consortium maintains a tight grip over the standard to provide manufacturers and software developers strict guidelines on the use and implementation of OpenGL. The standard will evolve to take advantage of more powerful and featured display adapters as they are developed, whilst maintaining a good degree of backward compatibility through software interpretation.

Controls

Having briefly discussed the basic principals behind OpenGL, the remainder of this section is devoted to the appropriate viewing controls available.

By default, the *Perspective view* initialises in **EDIT MODE**, which gives the same view of the object or scene as the three *Fixed views*. After using ***Imagine***, the last active *Perspective view* display will then be the one next initialised. As was discussed (starting on

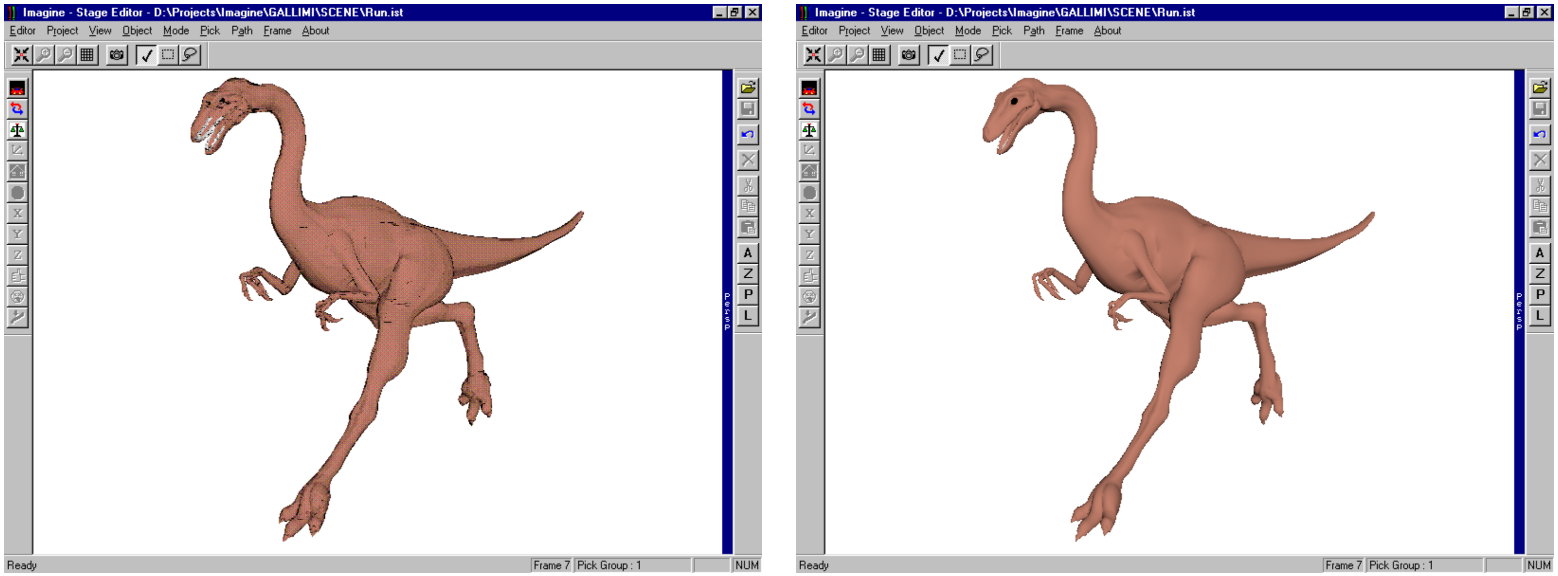


Figure 5: The highest quality Perspective view prior to OpenGL was the 216 colour flat shaded display (left). This remains an option for slower machines, whilst faster machines with or without a suitably enabled display adapter can benefit from OpenGL rendering (right).

page 1~15 of the I Files) there are various ways of "improving" the display in the *Perspective view* but the following has been added to accommodate OpenGL…

GL Use OpenGL ~ *View-Perspective menu* ~ Selecting this will, not surprisingly, render the *Perspective view* using the OpenGL routines, either through software or hardware interpretation. The OpenGL display mode is available to the **SOLID**, **SHADED** and **COLOR** display modes, all of which use hidden surface removal; a core routine within OpenGL. **EDIT MODE** and **WIREFRAME** are not supported, which is of little handicap as generally the overall surface geometry is of no concern when manipulating in these modes.

OpenGL can also be used in conjunction with the **SHOW LINES** (see page 1~16 of the I Files) option, which superimposes the visible edges on top of the OpenGL display.

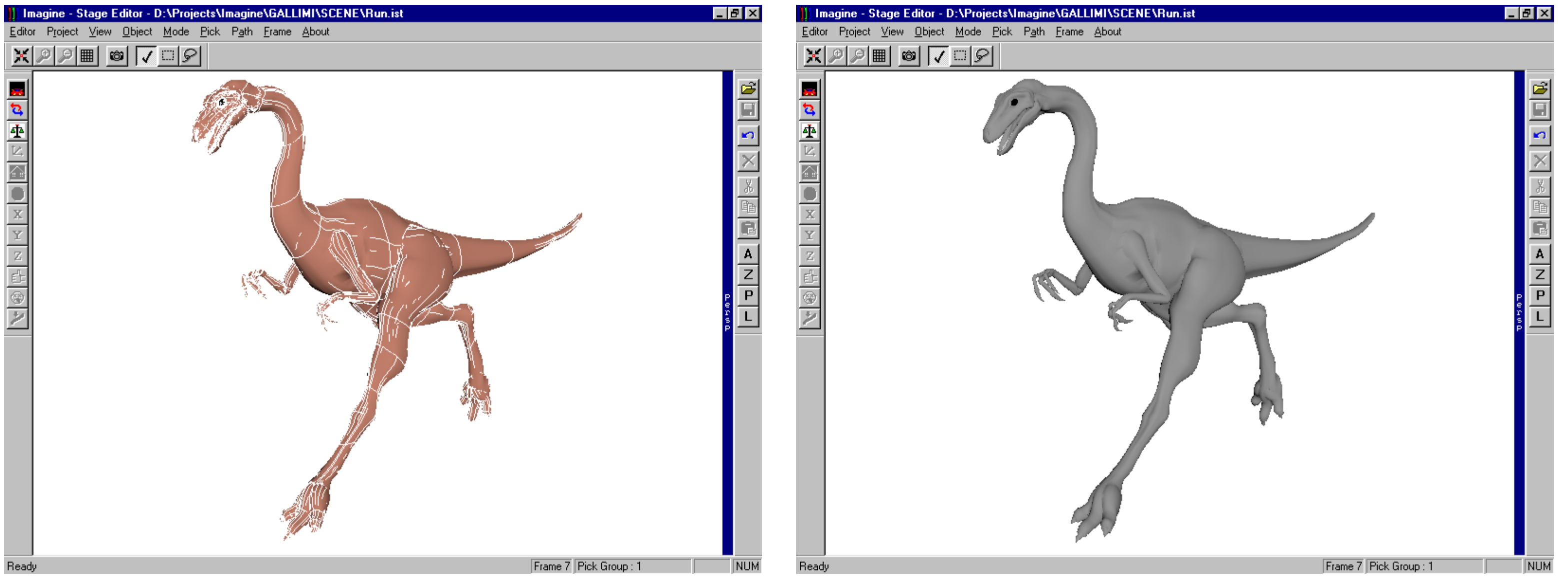
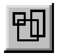


Figure 6: Where colour is unimportant or detracts from the scene, OpenGL can also be used with the Shaded option (right). The geometric complexity of an object can be guaged by using the Show Lines option in conjunction with the OpenGL display (left).

As can be seen in Figure 4, only the Base Color, Specular Color and Hardness — all taken from the current attribute settings — are interpreted under OpenGL.



Use Bounding Boxes ~ *View-Perspective menu* ~ Where the scene or object complexity adversely affects the responsiveness of the display, this option can be used in conjunction with the OpenGL display. This causes ***Imagine*** to update the display only according to the previously specified Bounding Box or Quick Edges as determined by the Quickdraw parameter in the *Attributes dialog* or the QUICKDRAW command (see page 1~30 in the I Files). These are implemented in the *Perspective view* when using the interactive ANGLE, ZOOM or PERSPECTIVE in all the *Editors* or when interactively manipulating the camera in the *Stage editor* whilst CAMERA VIEW is enabled.

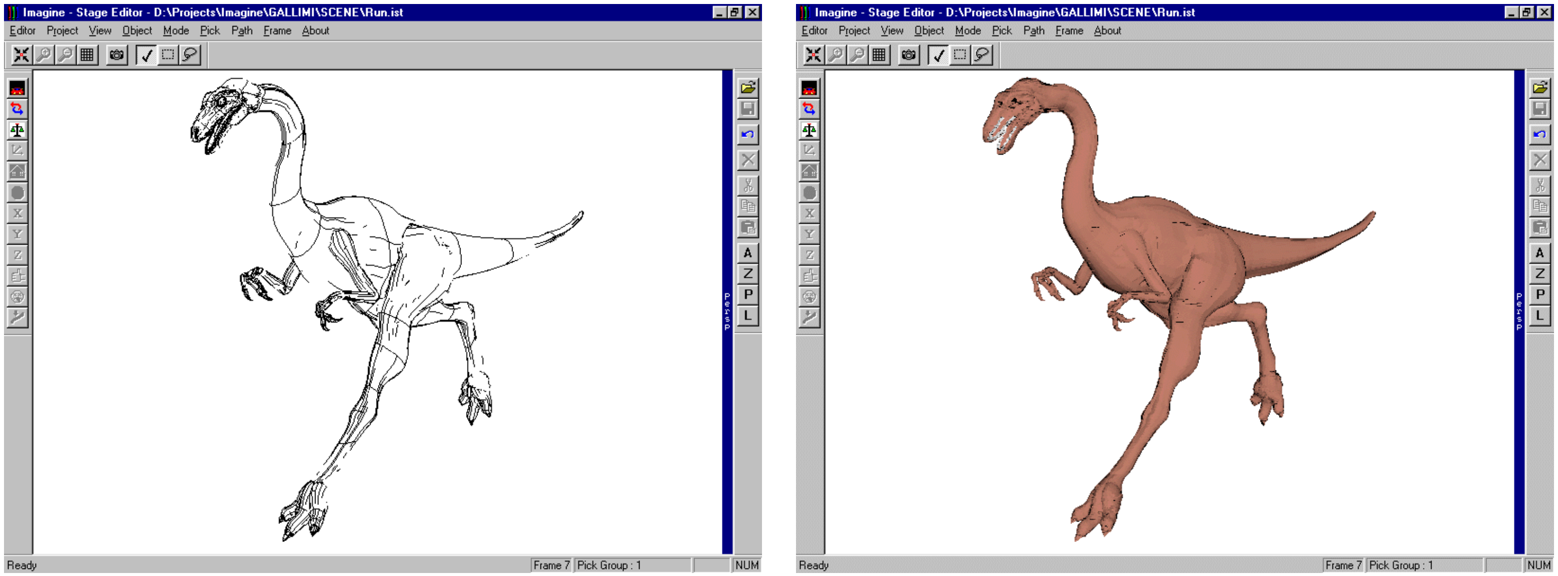


Figure 7: The use of OpenGL in the SOLID display mode will also increase the responsiveness of the display with a suitably equipped graphics adapter (left). Where hardware acceleration is not available, complex objects and scenes may be more responsive when using the 24 Bit/16 M Colors option.

Where hardware acceleration is available on the display adapter, the speed at which the *Perspective view* (and the Render window in the *Portals*) is refreshed will be considerably greater than when using software interpretation only. To improve the responsiveness of the display in these situations, but enhance the existing COLOR option, the following display type has been introduced…

24 Bit/16M Colors ~ *View-Perspective menu* ~ The new 24bit/16M COLORS display mode is a refinement of the previous 216 shades of grey/colour mode. Where OpenGL hardware acceleration is not available through an appropriate display adapter, the display in the *Perspective view* will often be more responsive and update quicker using this selection rather than the OPENGL option.

This option refers to how ***Imagine*** will compute the view internally. If the operating system has been set to a lower colour resolution, then of course, this will determine the final display.

Ini file

The following two parameters in the “Imagine.ini” file determines whether OpenGL or 24bit/16M Colors should be active or inactive the next time ***Imagine*** is started.

`[Editors]`

`UseOpenGL=0`

```
    0= OpenGL inactive
    1= OpenGL active
```

`Persp24Bit=0`

```
    0= 24bit/16M Colors inactive
    1= 24bit/16M Colors active
```

Aside from the restrictions mentioned earlier, OpenGL cannot render the objects according to any of ***Imagine***'s lights present in a scene; this includes both lights created in the *Detail editor* and those added in the *Stage editor*. This is as a consequence of the "local" rendering restriction referred to previously. OpenGL traditionally uses a "built-in" light, similar to ***Imagine***'s *Perspective view* light used when the SHADED or COLOR display modes are active. This, to a certain extent, restricts how the objects or scene can be displayed; a light source directly in front of the object may not reveal enough of the geometric detail as may be required. To improve this situation, the following simple but effective feature has been implemented…

interactive Lighting

As has been eluded to, the "front-on" lighting type may not illuminate an object in such a way as to be conducive to the modelling process. Whilst OpenGL normally accommodates only this simple lighting type, it does allow for extensions. ***Imagine*** now includes a variable light source — variable in position — for the *Perspective view* display and the newly introduced *Portal* Render window. This interactive Lighting control is also available to the new 24 Bit/16M COLOR and the older 216 COLOR display modes.

interactive Lighting ~ Control over the lighting is, as the name implies, interactive in that a new modifier button (radio button in the *Portals*) has been provided. Operation of the "L" button is similar to the other three view modifiers. Once active and when the cursor is anywhere within the confines of the *Perspective view* (and Render window in the *Portals*) ***Imagine*** interprets each click with the left mouse button as a new position for the light source. The same is true in the *Portals* but the Lighting Angle radio button has to be selected in preference to the Zoom or Viewing Angle options.

In the default configuration of ***Imagine***, the "L" button can be found to the right of the *Perspective view* directly below the A, Z and P buttons. These as you may recall, interactively control the angle, zoom level and perspective convergence of the *View*'s contents.

In the *Portals*, the lighting control can be found to the right of the Render window and below the other view modifier option buttons.

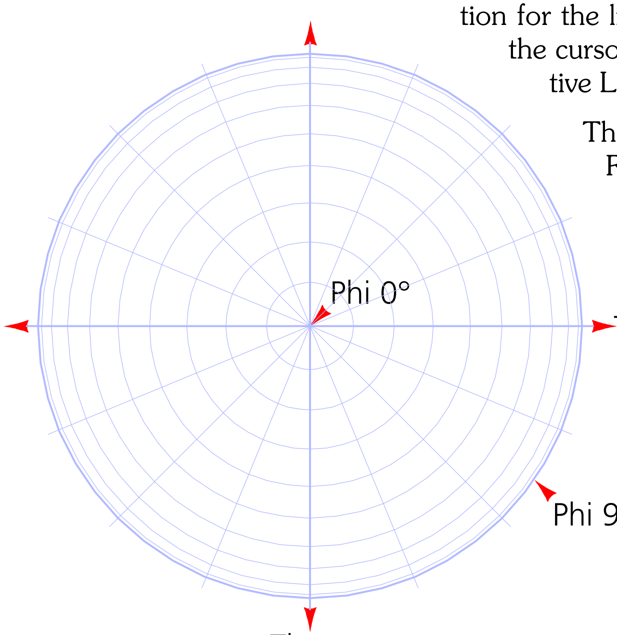
Click in the centre of the *View* and the lighting will stay the same, as this is the default position for the pseudo light source, in line with the *View*/Window camera.

Positioning

Whilst the lighting modifier is active, positioning the cursor over the *Perspective view/ Portal* Render window and selecting the left mouse button will determine the new position for the light source. Understanding how ***Imagine*** interprets the position of the cursor in the View/Window will allow more precise placing of the interactive Light source in such a way as to illuminate the object to the best effect.

The easiest way to visualise this is to imagine that the *Perspective view/* Render window is looking directly down at the “North Pole” of a lighting hemisphere — regardless of its actual orientation. The lines of latitude form circles centred on the view and extend out to its extremities. Longitudinal lines radiate out from the centre of the view. ***Imagine*** uses these two lines to describe a co-ordinate in three-dimensional space; anywhere on the surface of the hemisphere (see Figure: 8).

The hemisphere extends from the “North Pole” (directly inline with the camera) to the “Equator” which circles the centre of the *View*/Window — the *Perspective view* centre being determined by the three *Fixed views*.



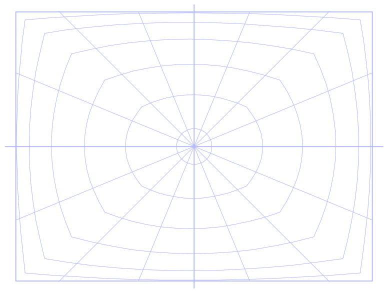


Figure 8: To transform two-dimensional co-ordinates into a three-dimensional lighting position, the Perspective view is treated as though a hemisphere is located in it (left). The “Equator” is centred on a position determined by the centres of the three Fixed views and the “North Pole” is directly inline with the perspective camera. The grid on the right shows how the extremities of the Perspective view lie on the “Equator”.

It should be borne in mind that the lines of latitude are in fact rectangles. This enables any point on the extremity of the *Perspective view*/ Render window to be interpreted as lying on the "Equator", 90-degrees from the "North Pole".

These circles of latitude (referred to in the "Imagine.ini" file as Phi; see below) allow the cursor to determine a point anywhere on the hemisphere from the "North Pole" to the "Equator". Thus, the Phi value determines the angle of departure from "straight-on" lighting, or the number of degrees from the "North Pole". Possible values run from 0-degrees (straight on) to 90-degrees (side lighting). Values of 91 to 180 degrees, which are not possible to interactively achieve, can be specified from the "Imagine.ini" file. These values will illuminate the object from behind; 180-degrees will position the light source in line with the "South Pole" of the hemisphere. Furthermore, with regard to the *Perspective view*, objects that are forward of the centre defined by the three *Fixed views* will also be increasingly back lit as the position of the interactive light effectively moves towards the "Equator".

Where the Phi value is greater than 0-degrees, the position according to the longitudinal lines (referred to in the"Imagine. ini" file as Theta; see below) is also taken into consideration. Achievable values for Theta extend from -180-degrees to 180-degrees, where:

> 0-degrees is a point anywhere along a horizontal line from the centre to the right side of the *View*/Window,
>
> 90-degrees is a point anywhere along a vertical line from the centre to the top of the *View*/Window,
>
> -90-degrees is a point along a vertical line from the centre to the bottom of the *View*/Window,

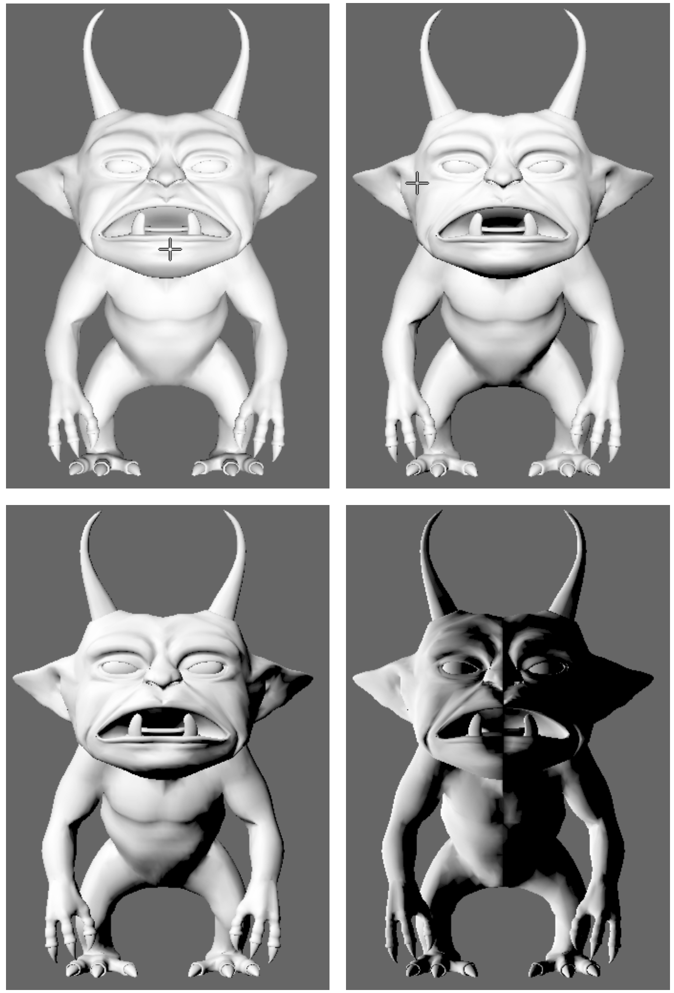


Figure 9: Varying the position of the interactive Light can reveal areas of detail to assist with the modelling process; 0,0 (top left) 22.5, 112.5 (top right) 45, 135 (bottom left) and 90, 180 (bottom right). The values described are taken from the "Imagine.ini" file and relate to the cursor positions: centre, up and left of centre, top left and centre left respectively. Troll object courtesy of Bill Fleming at Komodo Studios.

The position of the interactive Light not only determines how objects in the Perspective view are illuminated but can also determine the position of the Quickrender light.

Added towards the end of the "Constant Upgrade Programme" are two options allowing the Quickrender light to initially be placed according to the position of the interactive Light in the Perspective view or the previous Horizontal/Vertical Lighting Angle values in the Lighting tab of the Quickrender Settings dialog.

Selected by default, the "Initialize to match persective lighting" option sets the Horizontal/Vertical lighting angles to match those active in the Perspective view. Selecting the "Initialise lighting angles from previous values" uses the values active when the dialog was last used. Switching between these two options displays a dialog asking if the values should be applied immediately.

Whatever option is selected, the Horizontal/Vertical lighting angles values can be varied independently.

+/- 180-degrees is a point along a horizontal line from the centre to the left side of the *View*/Window.

When editing the "Imagine.ini" file 0 - 360-degree values can also be used.

The combination of these two values allows the two-dimensional input to be interpreted as any position on the surface of the three-dimensional lighting hemisphere.

The default position of the interactive Light is directly in line with the *Perspective view/Portal* Render window camera, in other words at the "North Pole" (0-degrees). However, the current position of the interactive Light is stored in the "Imagine.ini" file when ***Imagine*** is closed, thus maintaining that position when ***Imagine*** is restarted. At present, there is no way to reset the position of the interactive Light to its default 0-degrees other than manually editing the "Imagine.ini" file. Furthermore, the position of the interactive Light also determines the default value for the quickrender light when the *Quickrender Settings dialog* is opened.

Ini file

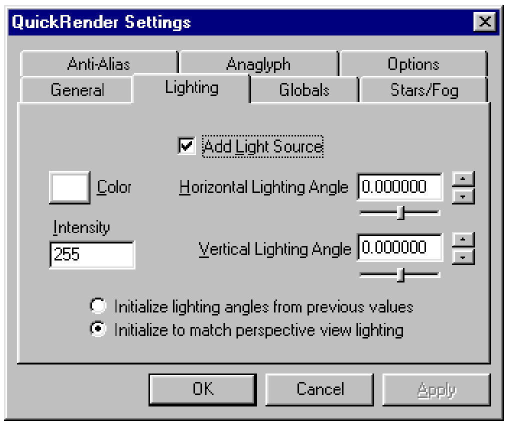
The following two values store the position of the interactive Light; this position being used the next time Imagine is started.

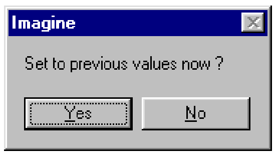
```
[Editors]
PerspLightPhi=0.000000
```

Initial "latitude" value for the interactive Lighting.

```
PerspLightTheta=0.000000
```

Initial "longitudinal" value for the interactive Lighting.





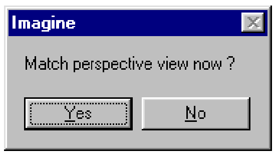


Figure 10: The Lighting tab in the Quickrender Settings dialog showing the two options determining the lighting angles to be initialy implemented.

The uses to which ***Imagine*** is put to are both wide and varied, ranging from the simplest of illustrations to complex animations. However a render is used there is often the need to combine part or all of it with another image; a process known as compositing. This is where two or more images are mixed together to create a seamless montage.

The creation and use of “simple” masks in most popular paint programs is one way to combine these various elements but can be time-consuming and produce artificial looking results when compared to using an…

Alpha channel

Certain image file formats are capable of storing 32 bits of information per pixel, that is 256 levels for each of the more familiar three colour channels, leaving room for an additional channel; usually referred to as the alpha channel.

This alpha channel can be used (by programs that support it) to regulate the application of the colour channels; red, green and blue. Without this alpha channel, each pixel comprising the image can only be displayed at the intensity determined by the values in the colour channels, producing the “familiar” colour image. However, by allowing the fourth channel to control the intensity at which the colour channels are applied, their strength can be varied. In other words, the alpha channel operates as a transparency mask, regulating the intensity of the pixel colours.

This alpha transparency mask is, as has been discussed, composed of 8 bits of information allowing each and every pixel to be one of 256 levels, where black is transparent, white is opaque and the intermediate levels of grey correspond to varying degrees of transparency. The darker the grey the more transparent the alpha channel is and the lighter the grey the more opaque. A mid grey with a value of 127 produces an alpha value with 50% transparency/opacity. Consequently, where an alpha channel is used to control the mixing of a foreground image with a background one, pixels in the foreground that correspond with black in the alpha channel will be fully transparent when mixed with a background image. Pixels that correspond with white in the alpha channel will be fully

Whilst there is no absolute standard, it is generally accepted that an alpha value of 0 (black) is treated as totally transparent and a value of 255 (white) fully opaque.

opaque and completely obscure the background. Whilst pixels in the foreground image that correspond with intermediate greys in the alpha channel will be partially transparent and produce an appropriate mix between the foreground and background images.

In short, each pixel in the foreground image is mixed with it's counterpart in the background image based on the alpha value for that pixel.

As well as allowing montages of elements from different images to be built-up, such things as animated characters, logos, text and image wipes can be composited with a video layer. Any variations in transparency across the individual elements will modify the underlying image appropriately. This then, is a key difference between using a "simple" mask to isolate an area within an image and using an alpha channel. Generally, a mask can only surround an element, any part of the image within the mask will completely obscure the underlying image. Of course, the edge of the mask can be "feathered" to blend the selection more effectively and the entire selection can be blended with the background at less than full strength, i.e. mixed with partial overall transparency. However, an alpha channel is capable of varying the transparency on a pixel by pixel basis.

Any number of images with their accompanying alpha channels can be layered on top of one another, enabling elements within the final image to be varied without having to resort to re-rendering the entire scene. Furthermore, the alpha channel — being just a greyscale layer — can be edited within most paint packages.

Having discussed what an alpha channel is and how it works, it is time to look at what ***Imagine*** looks for in a scene to generate an alpha channel.

Scene requirements

Fundamentally, ***Imagine*** looks for areas of transparency within a scene on which to base the alpha channel. The overall transparency of the scene is considered in relation to the background, whether it is a backdrop image, a star field or the horizon and zenith colours. This means that where the background is visible (from the camera's perspective) "around" the object's extremities, these areas are treated as transparent for the purpose of the generated alpha channel. Consequently, if the scene is in front or placed on top of an opaque "screen" (a ground plane for example) and the camera angle is such that the background cannot be seen, then the generated alpha channel would reflect this. The resulting alpha channel would be entirely white, i.e. opaque. The assumption being that since elements from the rendered scene completely fill the rendered image, then this scene would not be used as a foreground image; individual objects or object groups primarily forming this part of the composition. This, of course, assumes that the background is not visible through transparent objects.

Additionally, the amount of transparency (derived from Filter and Fog attribute settings) exhibited by each object directly influences the generated alpha channel.

Having discussed what ***Imagine*** is looking for in a scene, there are some important technical aspects that need to be considered to predict the alpha channel that will be generated and how it should be subsequently used.

Alpha channel technicalities

Imagine has always used three separate alpha channels internally, one for each of the red, green and blue colour channels to produce a "normal" render. These are used to accurately overlay the rendered scene on the background, whether it is a backdrop image, a star field or the horizon and zenith colours. Besides being used for determining the anti-aliasing for the edges of the objects against this background, it is also necessary to accurately mix objects with coloured filter and coloured fog attributes with themselves as well as any background present in the scene.

However, when the Generate Alpha Channel Image option is selected the final mixing between the three colour channels with the background in accordance with their associated alpha channels does not take place. Instead, the rendered scene is assumed to be overlaid against a black background to produce what is known as a standard "pre-multiplied" RGBA data.

Because the common image file formats only support a single alpha channel, this does pose the problem of how to combine the three that ***Imagine*** generates into one. With no obvious way of combining the three it uses the alpha channel with the largest values, on the assumption that anything less would not conform to the “pre-multiplied” concept. There is of course a certain amount of trade-off stemming from this, which is the main reason why, where possible, any composition should be achieved within ***Imagine***.

Without delving too deeply into the technicalities, it should be stressed that ***Imagine*** produces standard “pre-multiplied” alpha channels as opposed to the alternative “straight” variety. Both varieties provide the same control when compositing but if the wrong one is specified when mixing the foreground and background images in an image manipulation program, a loss of image quality can occur or colour inaccuracies in the form of “fringes” around objects become visible.

Whilst a “straight” alpha channel can produce better colour precision by keeping the transparency information strictly within the alpha channel, the “pre-multiplied” version is compatible with a wider range of programs. Additionally, when a “pre-multiplied” alpha channel is calculated with a known background colour it can be “un-multiplied” to create the “straight” alternative. With the “pre-multiplied” variety the transparency levels are not only stored in the alpha channel but also in the colour channels as the original background is replaced with black. It is this substitution of the existing backdrop image, star field, or horizon and zenith colours that produces the characteristic black background in “pre-multiplied” renders with an alpha channel.

With a “pre-multiplied” alpha channel, the background (backdrop image, star field or horizon and zenith colours) is replaced with a black background. This has to be so because multiplying by 0 (black) will ensure that the colour channel values do not exceed the alpha value; a prerequisite for the “pre-multiplied” variety.

The idea of a “pre-multiplied” alpha channel is really a convenience for image manipulation programs because the colour values should be less than or equal to the corresponding alpha value for each pixel, leading to a simple multiplication of the colour values in the original image by the alpha value divided by 255. In other words a “pre-multiplied” alpha channel acts as a mixing ratio between the foreground and background images. This is why colour from the background image should never appear in the “pre-multiplied” RGBA data for the foreground image. If it did, then the anti-aliasing around the object borders would contain colour and produce the “fringe” effects spoken of earlier.

One final observation is what happens in the alpha channel when the background (backdrop, star field or horizon and zenith colours) is visible through refractive objects. In scan-line rendering this presents no problems as the “rays” pass directly through refractive objects and onto the background with no deviation. Consequently, the alpha value reflects any variation in transparency across an object thus allowing a substitute background to be acceptably mixed at a later stage.

However, during ray-tracing “rays” that pass through transparent objects are deflected where the Index of Refraction is greater than 1. This of course presents a problem because it is impossible to store a transparency map for a distorted, unknown background image that will be composited later.

The solution could have been to render the alpha channel in the same way as it would have been in scan-line, but this would have caused visual confusion where other objects are visible through the refractive object. They would have been distorted by refraction whereas the background would not.

Alternatively, the ***Imagine*** background could have been falsely mapped to an imaginary plane placed within the scene but then the question of where in the scene should the plane be placed would arise. In addition, if the refractive object were a glass ball the expectation would be to see this plane warped.

Consequently, ***Imagine*** ignores the background completely and specifies no mixing with refractive objects with an Index of Refraction greater than 1, simply using the background colour in the case of a Quickrender or the horizon and zenith colours in the case of Renders for the staged project. A similar situation applies to reflective objects. There is of course, the option to specify an Index of Refraction of 1 where other requirements dictate the use of ray-tracing to produce the final render.

In the end, better results will always be derived from compositing within Imagine. However, where the intended background is unknown, ***Imagine*** produces an image that will mix in the best way possible within the obvious limitations.

Controls

Having discussed what an alpha channel is, the technicalities of what ***Imagine*** looks for in a scene and how it handles it, there is just a single check box to select in order that an alpha channel is generated…

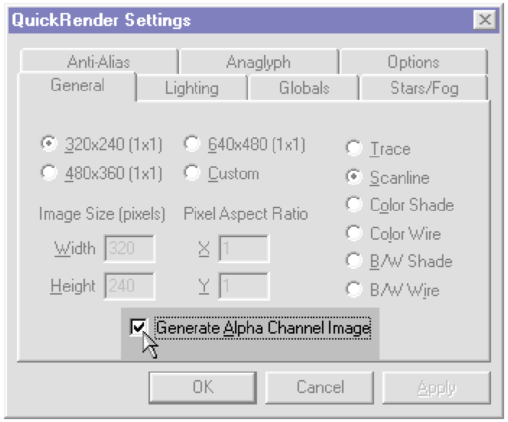


Figure 11: The Quickrender Settings dialog showing the Generate Alpha Channel Image option.

☑ **Generate Alpha Channel Image** check box ~ The Generate Alpha Channel Image option — available from the *General tab* of the *Quickrender Settings* or *Render Setup dialogs* — produces a rendered image with its associated alpha channel. This option causes ***Imagine*** to "mix" the rendered scene with a black background (rather than the background specified in ***Imagine***) to produce the "pre-multiplied" alpha channel data.

Once the Generate Alpha Channel Image option has been selected it is possible to view the alpha channel as it is generated, or once the render has been completed. By "right-clicking" over the *Render Output dialog* a pop-up menu is displayed. Along with the familiar controls for minimising, maximising and closing what is a "standard" operating system window and the more specific zoom commands, two new options have been included for viewing the alpha channel…

RGB Channel menu item ~ Selected by default, the RGB Channel selection displays the render with the three colour channels merged to give a full colour image (depending on the colour depth the operating system is set to).

When the alpha channel is being displayed (see below) selecting the RGB Channel option will display the render in its more familiar full colour form.

When producing "normal" renders with the Generate Alpha Channel Image deselected none of these items are available from the pop-up menu; only the three traditional colours channels will have been generated.

Alpha Channel menu item ~ Selecting the Alpha Channel option displays just the alpha data for the rendered image. Since alpha data is just a greyscale channel, the display will be composed of shades of grey where transparency varies across objects or black and white for opaque objects (but with anti-aliased edges).

Gamma Corrected sub-menu item ~ Whilst current thinking excludes the possibility of gamma correcting alpha channel data, there is the problem of correctly displaying this information when the Gamma feature (see page9~8 in the I Files) is active.

Selecting the Gamma Corrected option attempts to display the alpha channel in its true form, without any of the “hue-shifting” that might otherwise be apparent.

Uncorrected sub-menu item ~ Traditionally, alpha channel data should never be gamma corrected. Selecting this option will display the alpha channel without any gamma correction. This may produce “hue-shifting” in renders where gamma correction is used in the rendering process.

Save As menu item ~ To avoid the possibility of selecting a picture file format that does not support the fourth channel the Save As command has been revised. Selecting this command from the pop-up menu displays only those picture file formats — in the File Type drop-down list within the *Output File dialog* — that are capable of storing four channels; the fourth channel being the alpha channel.

The four common picture file types are all appended with the 32 suffix indicating the capability of storing four channels. However, whilst the BMP file type is capable of storing this fourth channel not all programs recognise the additional channel.

Whilst the Save As command is available from the pop-up menu in the *Render Output dialog* generated with both the QUICKRENDER and RENDER commands, the *Image Files tab* in the *Render Setup dialog* also restricts the available File Types when the Generate Alpha Channel Image option has been selected.

By default the Generate Alpha Channel Image is deselected however, the following entry in the “Imagine.ini” file will determine whether it is selected or not the next time Imagine is started.

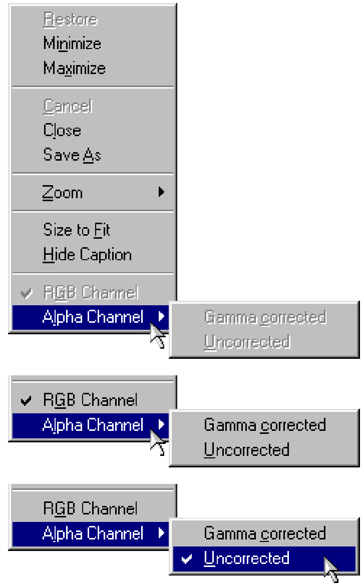


Figure 12: The pop-up menu displayed by right-clicking over the Render Output window showing the options available for viewing the alpha channel.

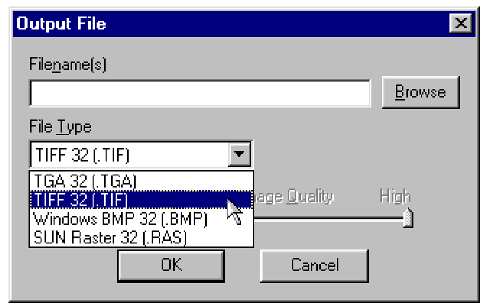
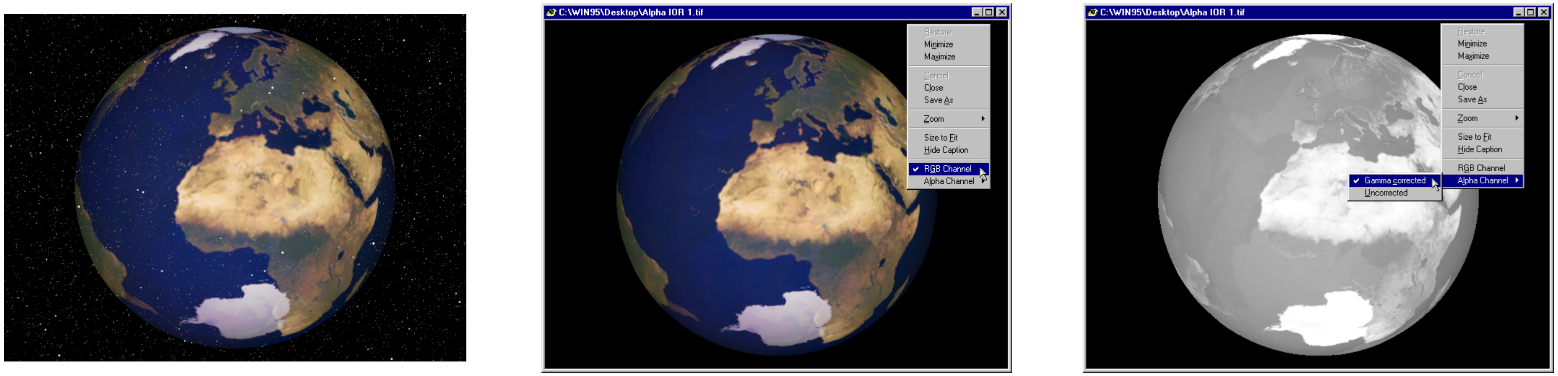
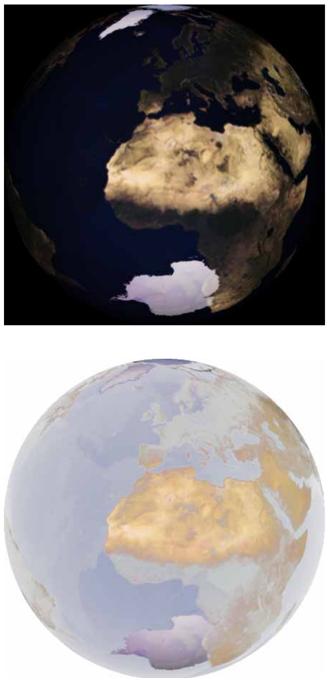


Figure 13: The Output File dialog showing how the File Types are restricted to only those that support 32 bits of information, i.e. those that can store an alpha channel.





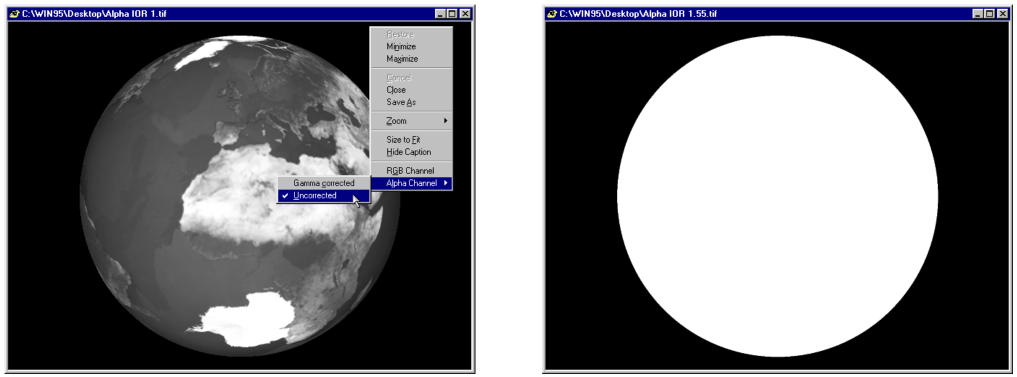


Figure 14: A simple example showing an image of the Earth filter mapped to a sphere with a starry background. A normal render (top left) a render with the Generate alpha channel option selected (top middle) viewing the generated alpha chennel gamma corrected (top right) viewing the alpha channel un-corrected (bottom middle) the alpha channel generated with ray-tracing where the sphere has an Index of Refraction greater than one (bottom right) and two views of the image according to the controlling alpha channel (bottom left).

Ini file

[Preferences]

GammaOff=0

> 0= The Generate Alpha Channel Image check box is not selected.
>
> 1= The Generate Alpha Channel Image check box is selected.

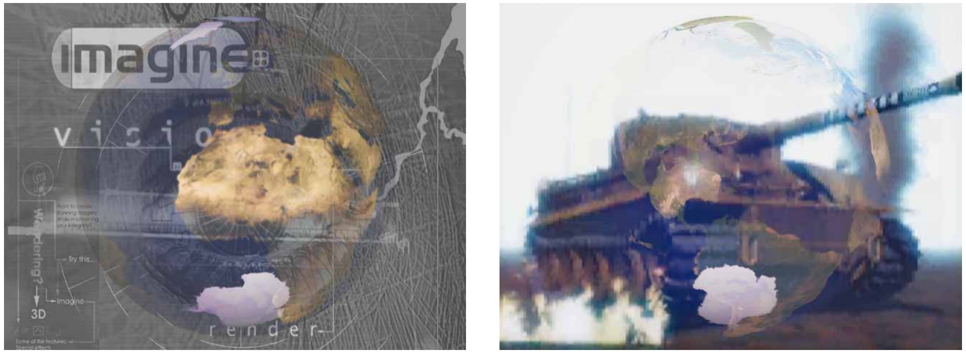


Figure 14: A couple of examples showing the globe composited with a background image using the associated alpha channel as a transparency mask (left) and against video footage (right). The video of the tank was produced by Phil Cook demonstrating how to animate tracked vehicles.

Transparency mapping

As an extension of the implementation of the Alpha Channel capability introduced during the Constant Upgrade Programme, the application of image maps can now be controlled by an alpha channel associated with the map.

As has just been discussed, an alpha channel associated to an image map controls the transparency of that map on a pixel by pixel basis. Ranging in values from 0 (black) which is fully transparent to 255 (white) which is fully opaque. This equates to transparent areas of the alpha channel allowing the full intensity of the brushmaps's RGB values to be applied, to opaque areas allowing none of the brushmap to be applied.

It is the individual pixel transparency control provided by an alpha channel that sets it apart from a brushmap's Mix/Morph parameter located on the *General tab* in the *Attributes portal* (see page 5~55 in the I Files). As you may recall, the Mix/Morph parameter in effect controls the transparency of the brushmap — or strength at which the map is applied — but it achieves this as a constant value applied to all the pixels across the entire image. Consequently, with a Mix/Morph value of 0.5, the image map will be applied with only half the intensity. The same would be true of an alpha channel composed of a flat mid grey with a value of 127.

A related function to the alpha channel is genlocking. This is where a specific colour is used to control the overlay of images. Prior to the introduction of the Alpha Channel feature the Use Genlock option (see page 5~77 in the I Files) allowed a specific colour in an image map to be ignored, i.e. be treated as transparent. This meant that any non-rectangular design could be mapped to an object by surrounding it with a specific colour. This colour was then ignored (or treated as transparent) and only the design applied. However, this method is inherently limited in that only a single colour value is treated as transparent and the edge between this and the design can appear "blocky" because any anti-aliasing is ignored. The use of higher resolution brushmaps will of course, minimise the "blocky" appearance. Additionally, anywhere on the design where the specified colour value exists will not be mapped onto the object.

Following the introduction of the alpha channel feature, which is more flexible than genlocking, the Use Genlock option on the *Wrapping tab* has been removed and a new *Tab* added to the Brushmaps section of the *Attributes portal*; namely the…

Transparency tab

The Transparency tab contains the following options controlling the transparency of the applied brushmap, from either an alpha channel or a single colour value…

- **Normal – no transparency** radio button ~ Selecting the Normal option applies the brushmap using the mapping type selected on the *Usage tab* according to the values contained in the red, green and blue colour channels, i.e. at full strength.

The only way to vary the strength at which the brushmap is applied whilst this option is selected is to use the Mix/Morph parameter on the *General tab*. Naturally, this controls the strength of application of the entire brushmap.

- **Use Alpha Channel Data (if available)** radio button ~ Selecting the Use Alpha Channel option will apply the brushmap according to the associated alpha channel data. This will control the strength at which the brushmap is applied regardless of the type of Full RGB mapping (excepting the Environment type). The application of brushmaps using the Red Channel Only Altitude mapping type cannot be controlled using the associated alpha channel data.

Where the alpha channel is black the corresponding colour channels will not be applied, whilst where it is white the brushmap will be applied at full strength. Values between these extremes (greys) will produce an appropriate variation in the strength of application.

⦿ **Use Transparent Color** radio button ~ The Transparent Color option replaces the older Use Genlock option mentioned earlier. This option allows a single colour value to be treated as transparent (not applied) wherever it appears in the applied brushmap.

Since this is a single value, the corresponding areas in the brushmap that relate to the chosen colour value will be treated as fully transparent and not applied. The strength of the application cannot be varied when using this option, other than resorting to the Mix/Morph parameter.

The main advantage of this option over its predecessor is that the transparent colour can be set specifically for each individual brushmap. The older Use Genlock option relied on the global setting found in the *Render Setup* and *Quickrender Settings dialogs* that determined the transparent colour to be used for all 24 bit brushmaps.

Transparent Color colour well ~ Selecting the colour well displays the standard operating system *Color dialog* from which the colour in the brushmap that is to be treated as transparent (not applied) can be specified.

The colour value can be numerically specified in either the RGB or HSL colour space models, or interactively picked using the mixing palette and the luminance slider.

❑ **View Alpha Channel** button ~ Selecting the View Alpha Channel button displays just the alpha channel for the selected brushmap (as a greyscale) in a standard operating system window.

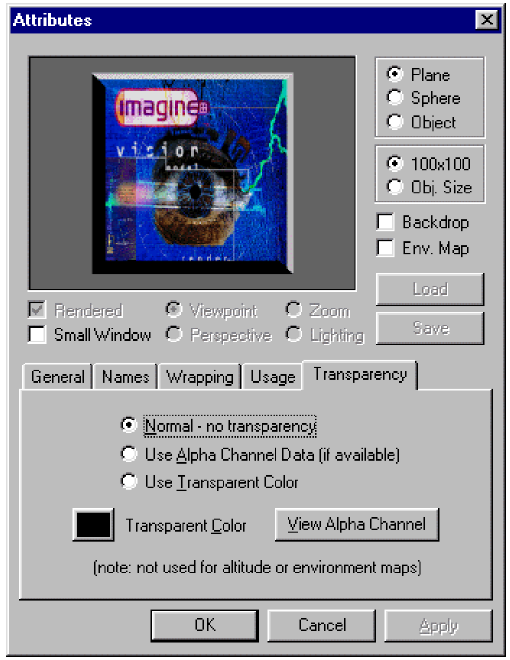


Figure 15: The Attributes portal showing the new Transparency tab with its default settings.

Selecting the right mouse button whilst the cursor is over this window will display the pop-up menu described earlier allowing, among other things, the colour image to be viewed.

Loading objects creating in previous versions of ***Imagine*** that had the Use Genlock option enabled for any applied brushmaps will now show the Use Transparent Colour option selected instead. Furthermore, the transparent colour for 24 bit brushmaps will be set to black (0,0,0) regardless of the colour value originally specified; the previous value if different to this will have to be reinstated.

When using 8 bit brushmaps the transparent colour can now be selected using the colour well. This is a departure from previous versions of Imagine where colour 0 in the 8 bit palette was always treated as transparent.

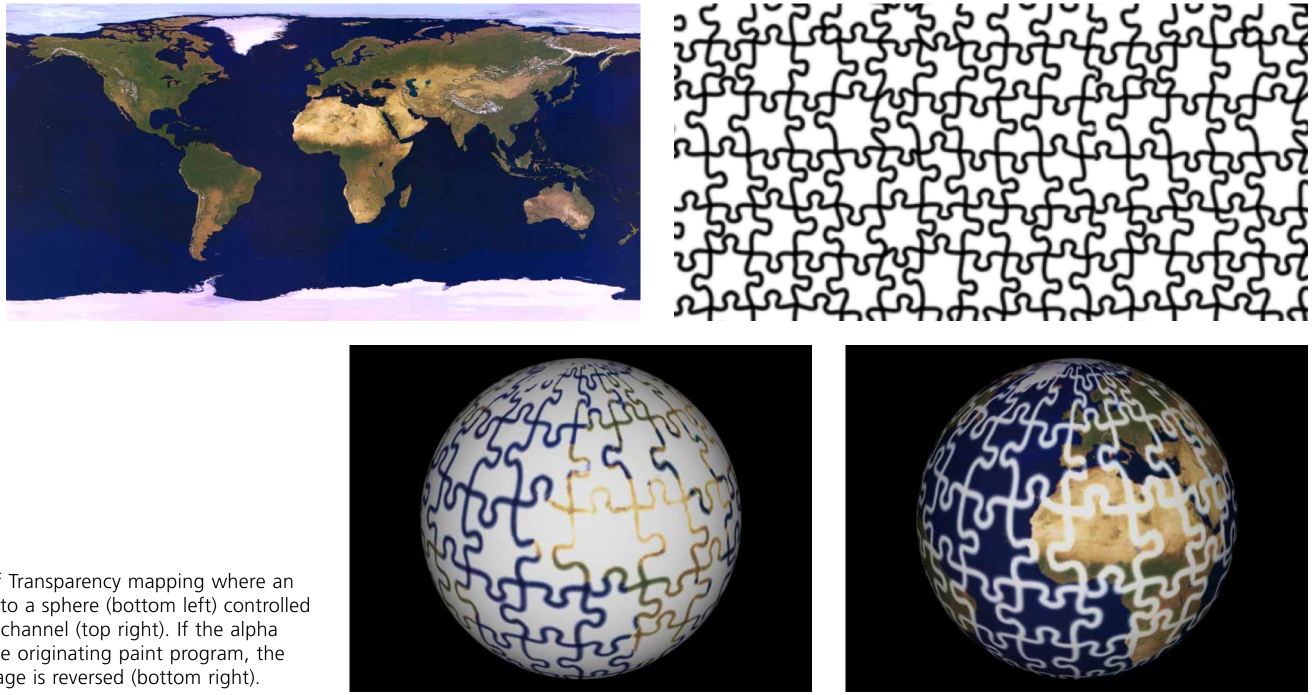


Figure 16: An example of Transparency mapping where an image (top left) is mapped to a sphere (bottom left) controlled by an associated alpha channel (top right). If the alpha channel is reversed in the originating paint program, the application of the image is reversed (bottom right).

In previous versions of ***Imagine***, rendering an image that did not conform to the standard 4:3 aspect ratio (VGA landscape format) or thereabouts, often proved problematic. This was because the display in the *Perspective view* dictated the "framing" of a scene for a Quickrender (or Render when CAMERA VIEW was enabled). Consequently, if the rendered image was not proportioned similarly to the *Perspective view* then either more of the scene than was required would have been rendered (thus wasting time and resources) or worse still, parts of the scene may have been omitted from the render.

Under such circumstances, it was strictly a trial and error process, the only guideline being that the render would include the full width of the *Perspective view*. For square or portrait format rendered images, the height value had to be initially guessed and adjusted following a test render.

The old trick of rotating the camera through 90-degrees on it's local Y-axis did give a more accurate indication for framing a portrait render. However, problems did arise, such as backdrop images would not be rotated and the reflection from global maps would appear rotated, which may have appeared odd. There was of course, the inevitable neck strain that had to be contended with, resulting from trying to manipulate the camera at such an awkward angle!

This somewhat grey area has been swept away with the introduction of the…

Render Frame

The RENDER FRAME command — alternatively known as the "safe area" or "render area" — highlights the area of the display in the *Perspective view* that will be included in the render according to the proportions specified in the *Render Frame Proportions dialog*.

There are two general circumstances under which RENDER FRAME is used and will influence how it is imposed. These are dependent on whether a QUICKRENDER or RENDER will be generated; a QUICKRENDER being based on the *Perspective view* display and a RENDER on what the camera is actually viewing.

As will be recalled, when a scene is quickrendered the display in the *Perspective view* determines what should be included in the render. With this as the scenario the Render Frame of the chosen proportions is scaled to fit the *Perspective view*. In other words, the Render Frame will be determined by the largest rectangle with the specified proportions that will “fit” in the *Perpsective view*. The **interactive ZOOM** currently in use for the *Perspective view* and the relative **ZOOM** in the three *Fixed views* in conjunction with the Render Frame will all contribute to framing the scene; this applies in all *Editors*.

In the *Stage editor* with the **CAMERA VIEW** option active (see page 7~7 in the I Files) the display in the *Perspective view* is determined by the position and orientation of the camera. In this instance, the horizontal aspect of the Render Frame is initially calculated according to the size of the camera, in other words its field of view (see page 7~46 in the I Files). To calculate the vertical aspect of the Render Frame, the vertical proportion value supplied in the *Render Frame Proportions dialog* is used in conjunction with the horizontal field of view of the camera to produce the specified Render Frame proportions.

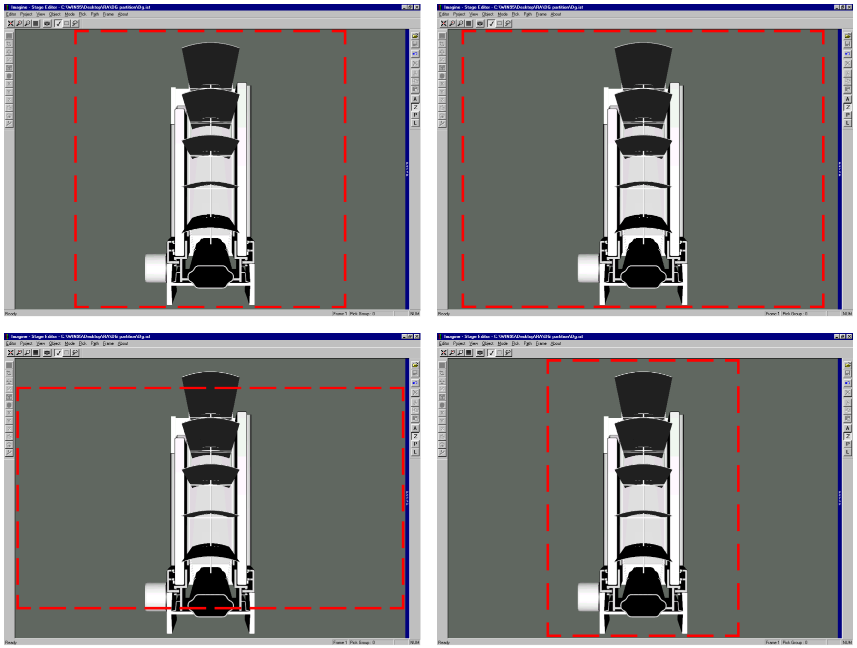


Figure 16: In all circumstances other than in the Stage editor with CAMERA VIEW enabled, the Render Frame will be determined by the largest rectangle with the specified proportions that will “fit” the current Perspective view.

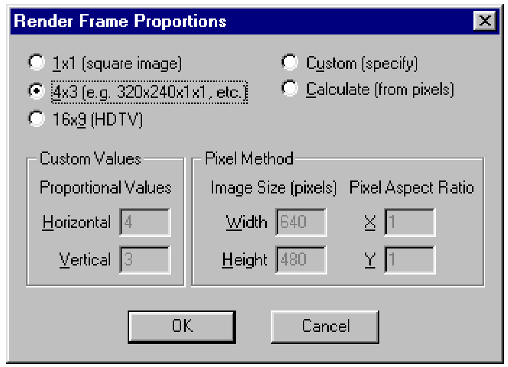


Figure 17: The Render Frame Proportions dialog showing the 4×3 default selection.

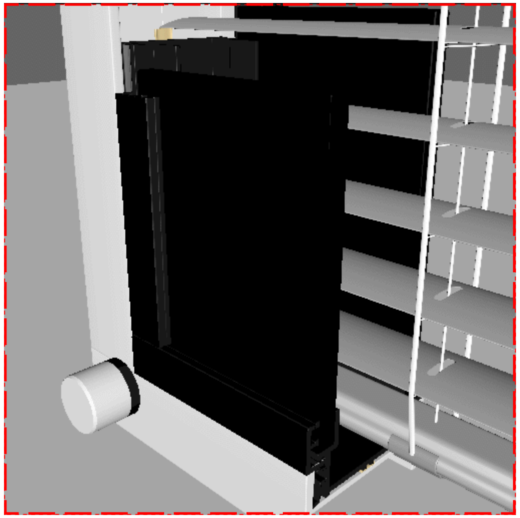
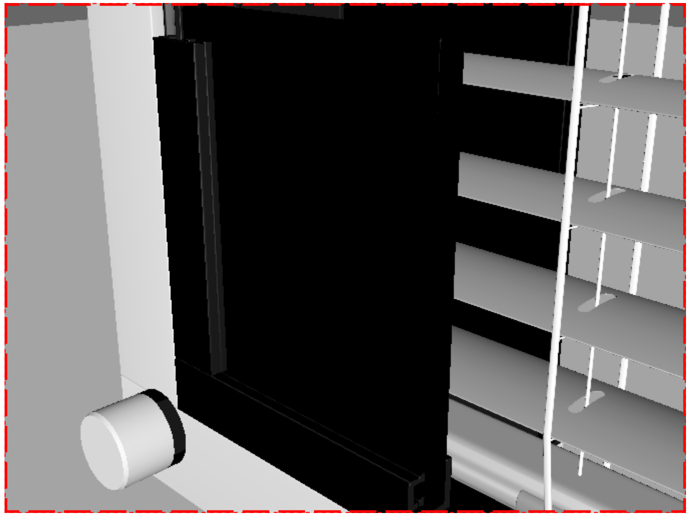
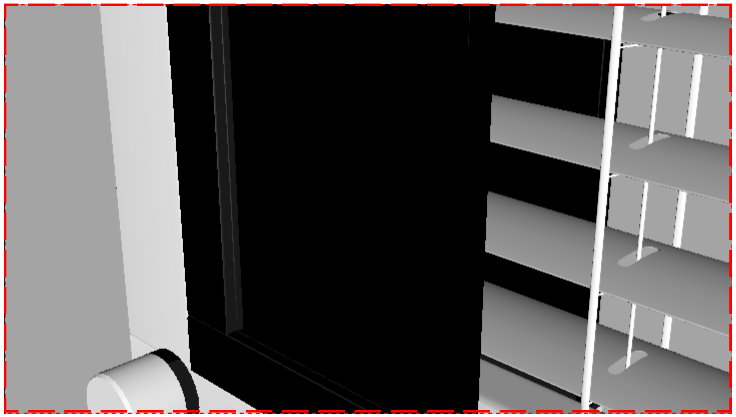
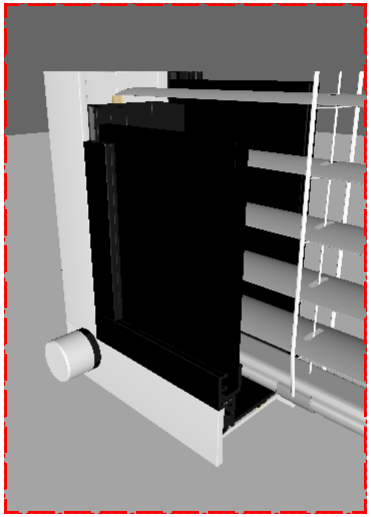
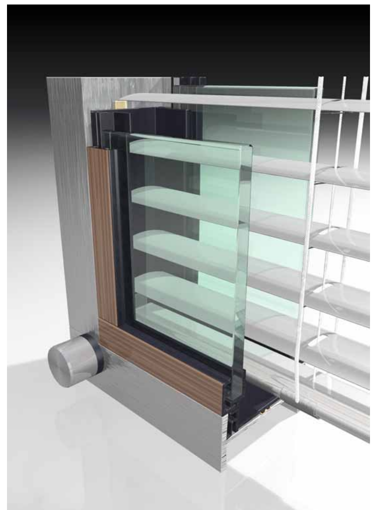






Figure 18: In the Stage editor with CAMERA VIEW enabled, the zoom ratio of the Perspective view may be adjusted to suit the Render Frame. In the staged scene shown the following proportions were specified: 1×1 (top left) 4×3 (top right) 16×9 (bottom left) 70×99 (bottom right). The 70×99 proportions were used to produced the A4 format render.

Having determined the proportions of the Render Frame it is sized to fit the *Perspective view* as before, but in this case the zoom ratio of the *Perspective view* itself may also be varied in order to view the entire Render Frame.

It should be noted that whilst the display in the *Perspective view* is scaled to fit the Render Frame, this does not affect the camera placement, alignment or size within the staged scene.

The correct use of the Render Frame does of course assume, that an image size specified in either the *Quickrender Settings* or *Render Setup dialogs* is proportionally identical. In other words, as long as the rendered image aspect ratio is the same as the Render Frame aspect ratio, then the highlighted area in the *Perspective view* will accurately reflect the eventual render.

Available from all *Editors*, the following pair of commands allow the Render Frame to be displayed and its proportions specified…

> **Show** ~ *View-Render Frame menu* ~ Deselected by default, selecting the Show command will superimpose the Render Frame on the Perspective view display, according to the current settings in the *Render Frame Proportions dialog* (see below). SHOW is a toggle command in that once active a tick appears next to the command; to remove the Render Zone display, select the command again.

The extent of the Render Frame is shown as a red dashed line describing a rectangle or square to highlight it against the *Perspective view* display.

> **Set** ~ *View-Render Frame menu* ~ Selecting the Set command displays a *Dialog* allowing the aspect ratio (proportions) of the Render Frame to be set.

The displayed *Render Frame Proportions dialog* contains the following parameters…

> ⦿ **1x1 (square image)** radio button ~ This option, not surprisingly, displays a Render Frame with a square aspect ratio, i.e. the height matches the width.

Having specified this option, the display in the *Perspective view* will be adjusted to show the extent of the scene that will be covered when rendering square images.

⊙ **4×3 (e.g. 320×240×1×1, etc.)** radio button ~ Selected by default, the 4×3 aspect ratio will frame the scene to almost fill the *Perspective view*, since it is virtually the same format.

This is the standard screen format for most common IBM compatible screen resolutions (320×240, 640×480, 800×600, etc). Additionally, both the NTSC and PAL video standards use this ratio for overscanned screens — 720×486 and 768×576 respectively — as does the Academy film format (2048×1536).

⊙ **16×9 (HDTV)** radio button ~ This option produces a Render Frame suitable for framing scenes that are destined for rendered images suitable for high definition television (1280×720 or 1920×1080).

⊙ **Custom (specify)** radio button ~ Where the aspect ratio of the rendered image does not conform to one of the foregoing three pre-sets, but is "known", selecting the Custom option allows the values to be supplied directly by activating the following input boxes…

Custom Values

Proportional Values ~

☐ **Horizontal** input box ~ The value supplied for the Horizontal parameter — expressed as a ratio — governs the "width" of the Render Frame.

This will be used in conjunction with the Vertical parameter (see below) to calculate the overall aspect ratio of the Render Frame.

☐ **Vertical** input box ~ The value supplied for the Vertical parameter — expressed as a ratio — governs the "height of the Render Frame.

This will be used in conjunction with the Horizontal parameter (see above) to calculate the overall aspect ratio of the Render Frame.

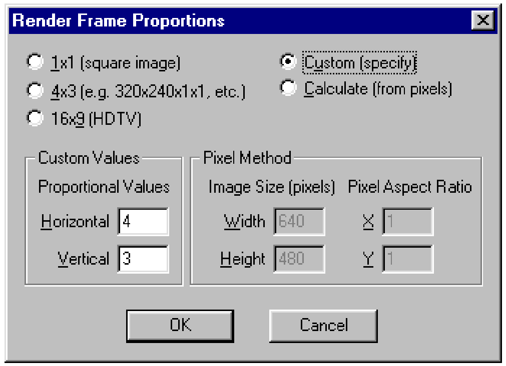


Figure 19: The Render Frame Proportions dialog showing the Custom option selected and the Custom Values section available.

⦿ **Calculate (from pixels)** radio button ~ Where the aspect ratio of the image to be rendered is “unknown”, selecting the Custom option allows the proportions of the Render Frame to be determined by pixel width and height values.

Specifying the pixel counts will allow ***Imagine*** to calculate the aspect ratio for the Render Frame. The results can be viewed in the Horizontal and Vertical input boxes within the Custom Values area of the *Render Frame Proportions dialog*, even though they are not “available” (see above).

The Calculate option also allows the proportions of the Render Frame to reflect non-square pixels. This allows the rendered image to be accurately framed according to any output device that requires the pixels themselves to be non-square.

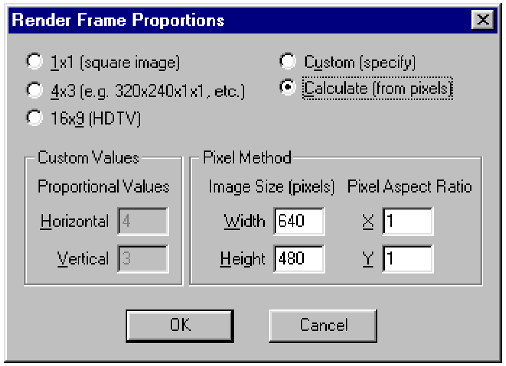


Figure 20: The Render Frame Proportions dialog showing the Calculate option selected and the Pixel Method section available.

Pixel Method

Image Size (pixels) ~

☐ **Width** input box ~ The value supplied for the Width parameter will determine the horizontal dimension of the Render Frame.

This will be used in conjunction with the Height parameter (see below) to calculate the overall proportions of the Render Frame.

☐ **Height** input box ~ The value supplied for the Height parameter will determine the vertical dimension of the Render Frame.

This will be used in conjunction with the Width parameter (see above) to calculate the overall proportions of the Render Frame.

Pixel Aspect Ratio ~

☐ **X** input box ~ The value supplied for the X parameter will govern the “width” (expressed as a ratio) of each pixel comprising the rendered image.

Used in conjunction with the Y parameter (see below) this will determine the aspect ratio of the pixels, i.e. whether they are square or rectangular in shape.

□ **Y** input box ~ The value supplied for the Y parameter will govern the "height" (expressed as a ratio) of each pixel comprising the rendered image.

Used in conjunction with the X parameter (see above) this will determine the aspect ratio of the pixels, i.e. whether they are square or rectangular in shape.

Ini file

The following is a list of entries in the "Imagine.ini" file, found in the Windows folder. Inclusion of an item within this file indicates that the particular item's values are stored and will be initialised the next time ***Imagine*** is opened.

[Editors]

`ShowFrame=0`

`0= The Show Frame feature is disabled.`

`1= The` Show Feature is enabled.

`RenderXAspect=4.000000`

`Initial width ratio for the displayed Render Frame.`

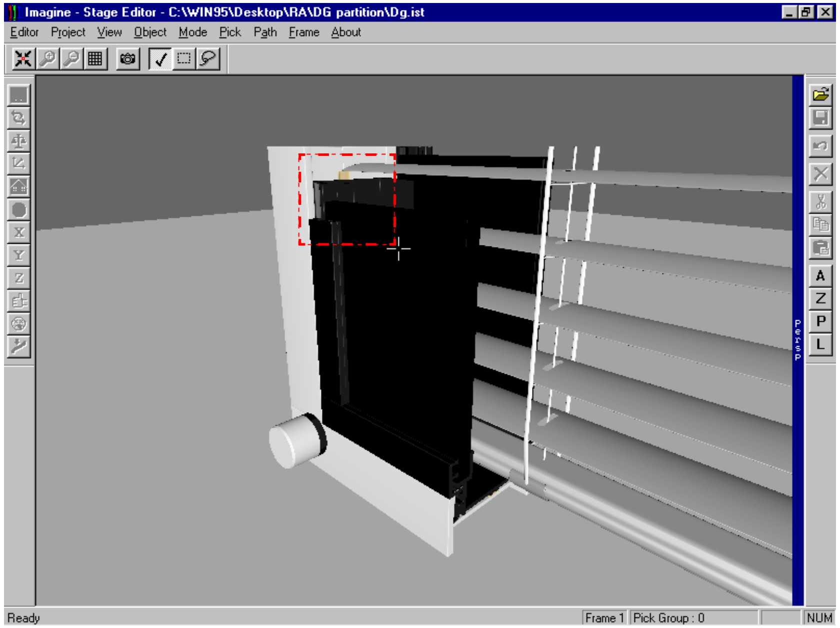
`RenderYAspect=3.000000`

`Initial height ratio for the displayed Render Frame.`

Set Zone

The ability to highlight an "important" area within the *Perspective view* has been extended to improve the functionality of another command; that of **SET ZONE** (see page 1~21 in the I Files).

Previously, the only way of identifying the area that had been defined as a limited render zone was to **RENDER** or **QUICKRENDER** it. Now, any area defined with the **SET ZONE** command is highlighted with a dashed red line. This not only acts as a visible reference in the *Perspective view* of the defined area, but also serves as a reminder that one is active. This alert is all the more apparent due the fact that the **SET ZONE** marquee cannot be switched off, even when the **SHOW** command is deselected. It has been designed this way to diminish the possibility of rendering a time-consuming animation to discover that only a portion of it has actually been rendered.



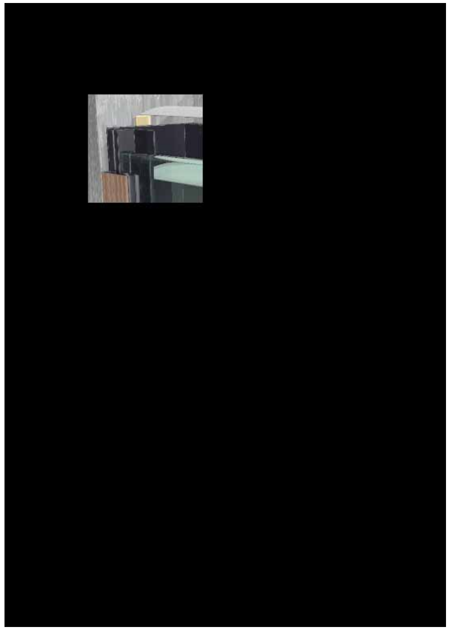


Figure 21: The Set Zone command displays a marquee in the Perspective view (left) highlighting the area that will be rendered (right).

No matter how proficient you become at controlling lighting and defining the surface attributes to produce realistic three-dimensional looking renders, they will always remain as two-dimensional images. ***Imagine*** offers another rendering feature that, whilst still producing a two-dimensional image (or animation) conveys the impression of "real" three-dimensional volume in a scene...

Anaglyph... 3D stereo rendering

Human vision is highly attuned to resolving the world around us in three-dimensions. This is partly due to the brain interpreting the images it receives from the left and right eye. Because the eyes produce separate images from slightly different viewpoints of the same "target", the brain is able to compile this information into a single image with accurate depth information - known as Stereoscopy. This stereo image processing gives what we see around us such depth. Just try to reach for something whilst one eye is covered!

Most of the time spent creating images and animations within ***Imagine*** is trying to recreate this three-dimensional illusion on a two-dimensional screen. However, using the eye/brain stereo image processing ability, the illusion of depth in an image or animation can be improved by partially replicating this ability. This can be achieved by producing an image that is a fusion of the scene being rendered from two slightly different viewpoints.

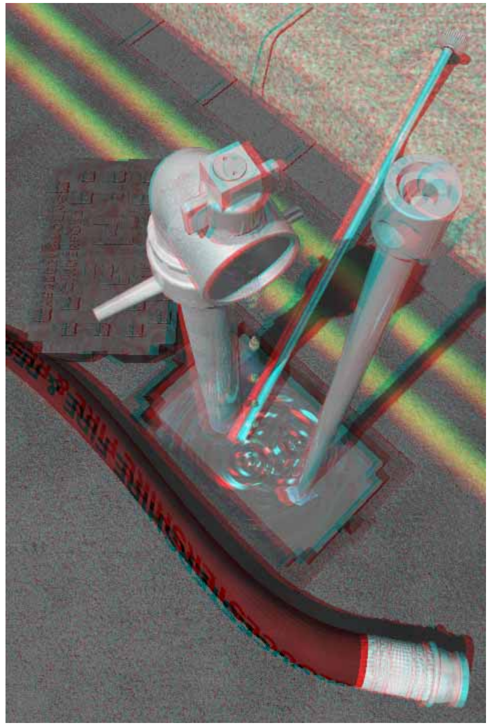


Figure 22: When viewed through Anaglyph glasses the colour Anaglyph on the right conveys real "volume" in a scene, something that a traditional render can never do (left)

Anaglyph is one method of producing apparently three-dimensional images from a collection of methods known as Stereograms. An Anaglyph 3D stereo image requires the user to wear red and blue/green lens glasses; this is considered the most effective method to experience stereoscopy on a computer.

Imagine will, depending on the parameters set, render the scene from two slightly different camera positions corresponding to the observers left and right eye. These two images are then “fused” into one composite image that uses colour to encode portions of the image that should only be visible to either the left or right eye when using Anaglyph glasses. This then replicates how the eyes and brain function together, but with a single image.

There are three ways to colour encode an Anaglyph 3D Stereo image, all of which are supported by ***Imagine***…

- Colour Anaglyphs try to preserve as much of the original colour information. Some scenes will not work as well as others producing varying degrees of retinal rivalry and ghosting, but generally the results can be quite spectacular.
- Greyscale Anaglyphs discard the colour information in a scene, but are usually easier to view and resolve the “depth” in an image more successfully.
- Pure Anaglyphs use just the red and blue (or green) colour information within a scene. These often give the best three-dimensional effect but again, sacrifice the colour information, producing only a greyscale image.

For a particular scene, try the colour option first. If the results are not as might be expected try greyscale, and if this is not acceptable use the pure option.

When viewing Anaglyph 3D stereo images with Anaglyph glasses, it is generally accepted that the red lens is placed over the left eye and the blue/green lens the right. This is in accordance with the International Stereoscopic Union.

If you experience eye fatigue or discomfort, remove the glasses and take a break. It is recommended that you only put the glasses on when you want to view an image.

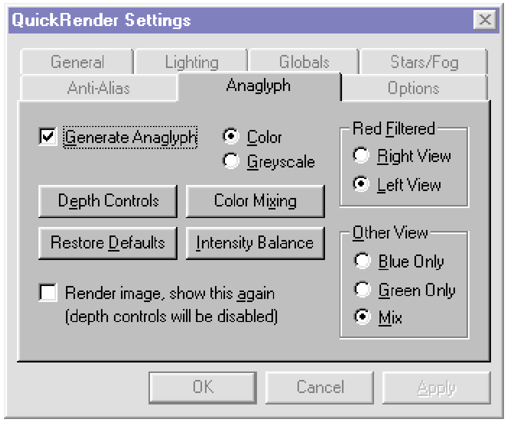


Figure 23: Once the Generate Anaglyph option is activated, all the other parameters become available.

Controls

The Anaglyph controls are all contained in the *Anaglyph tab* of the *Render Setup* and *Quickrender Settings dialogs*. Bear in mind that the *Anaglyph tab* in the *Render Setup dialog* will be unavailable until at least one frame has been specified in the *Frames tab*.

Initially, only three controls are available, the remainder being “ghosted-out”. In ***Imagine***’s default condition, Anaglyph rendering is disabled (the Generate Alpha option being de-selected) producing a “normally” rendered image.

☑ **Generate Anaglyph** check box ~ Selecting the Generate Alpha option allows an Anaglyph image to be rendered. The remaining options on the *Dialog* become available.

⦿ **Color Anaglyph** radio button ~ The Colored option allows full colour Anaglyph 3D stereo images or animations to be produced.

The colours in an Anaglyph image will not be as they appear in a normal render; the colour saturation has to be somewhat reduced to allow an Anaglyph image to be easily viewed. As will become apparent, attempting to maintain the colour saturation in the original image generally produces Anaglyphs that are not easily “fused” by the brain.

⦿ **Greyscale Anaglyph** radio button ~ The Greyscale option discards

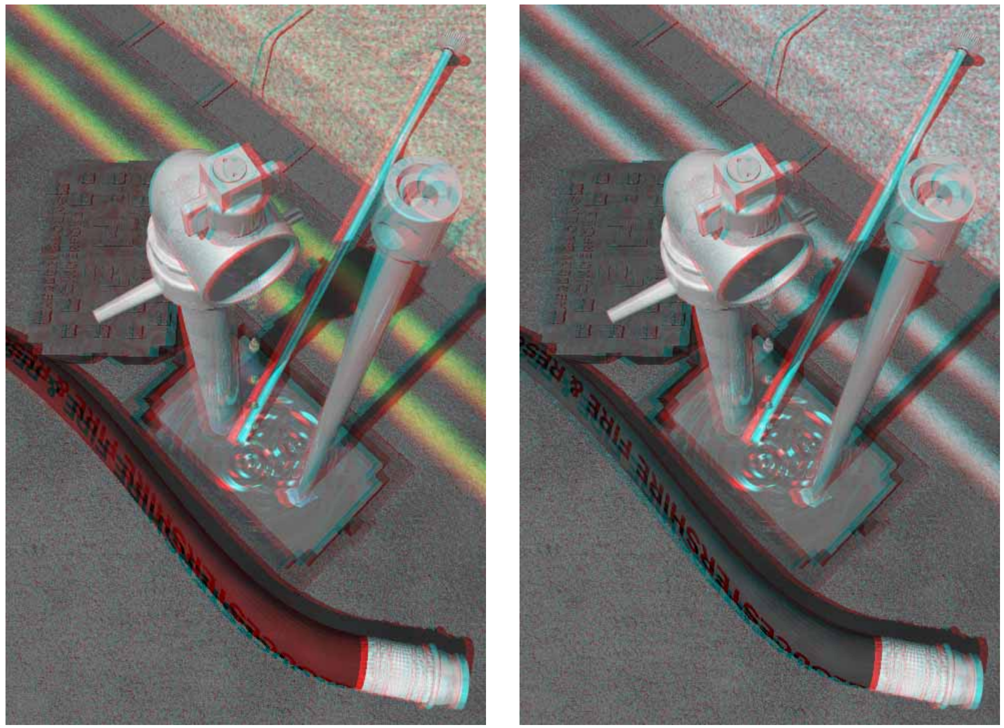


Figure 24: A greyscale Anaglyph (left) and a colour Anaglyph (right) derived from the default parameters. Notice that whilst the colour Anaglyph looks more “impressive” it is slightly more difficult to view.

all the colour information within the scene, producing an Anaglyph image or animation composed of shades of grey.

The image will of course, contain the specified colour encoding to produce the stereo effect, but when viewed through Anaglyph glasses the image will be portrayed as various shades of grey.

The effect of selecting the Color or Greyscale Anaglyph option initially appears to have no discernible result, in that the Color Mixing button remains the same. However, the controls revealed from this button are completely different; as will also become apparent.

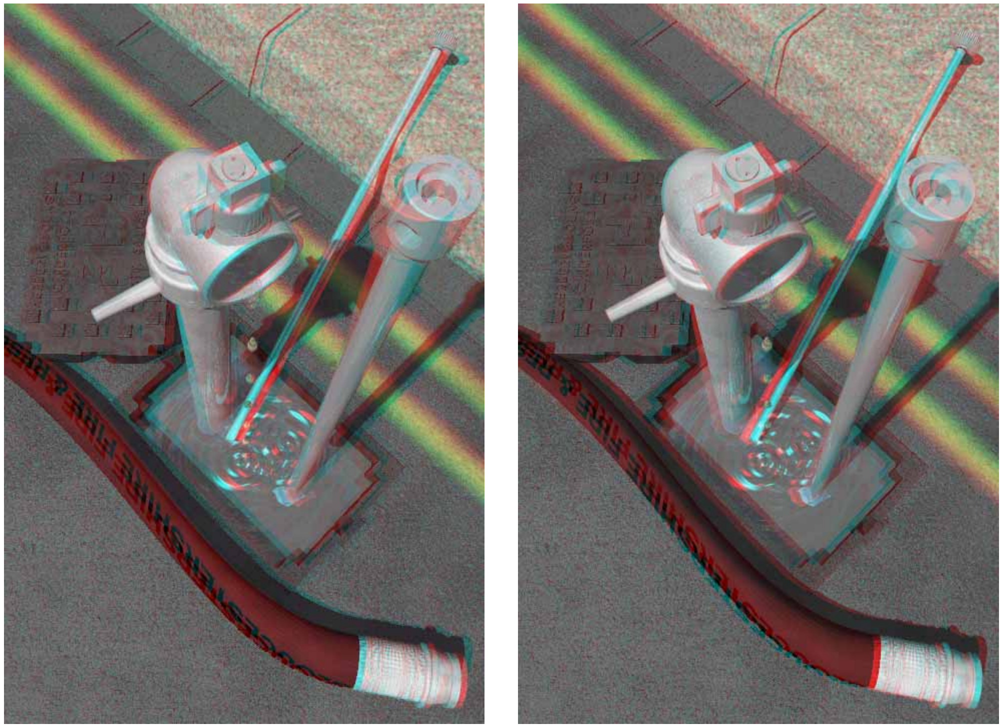


Figure 25: Traditionally the red lens is the left lens (right) but to cater for any Anaglyph glasses that have the red lens on the right there is the option to switch them (left).

Red Filtered:

⦿ **Right View** radio button ~ Selecting the Right View option colour encodes the Anaglyph image in such a way that the red colour component is destined for the right eye.

In other words, to view the Anaglyph successfully the glasses should have the red lens covering the right eye so that only red light from the image reaches it.

⦿ **Left View** radio button ~ Selecting the Left View option colour encodes the Anaglyph image in such a way that the red colour component is destined for the left eye. This is the default selection and as has already been said is the generally accepted way of encoding Anaglyph 3D stereo images.

In other words, to view the Anaglyph successfully the glasses should have the red lens covering the right eye so that only red light from the image reaches it.

Other View:

- **Blue Only** radio button ~ Selecting the Blue Only option allows pure Anaglyph images to be composed, where the Anaglyph glasses being used have a blue lens.

- **Green Only** radio button ~ Selecting the Green Only option allows pure Anaglyph images to be composed, where the Anaglyph glasses being used have a green lens.

Selecting either the Blue or Green options will automatically select the Greyscale Anaglyph option, if Color Anaglyph is active.

- **Mix** radio button ~ Selecting the Mix option allows full colour Anaglyphs (in addition to greyscale Anaglyphs) to be generated by combining both of the preceding options.

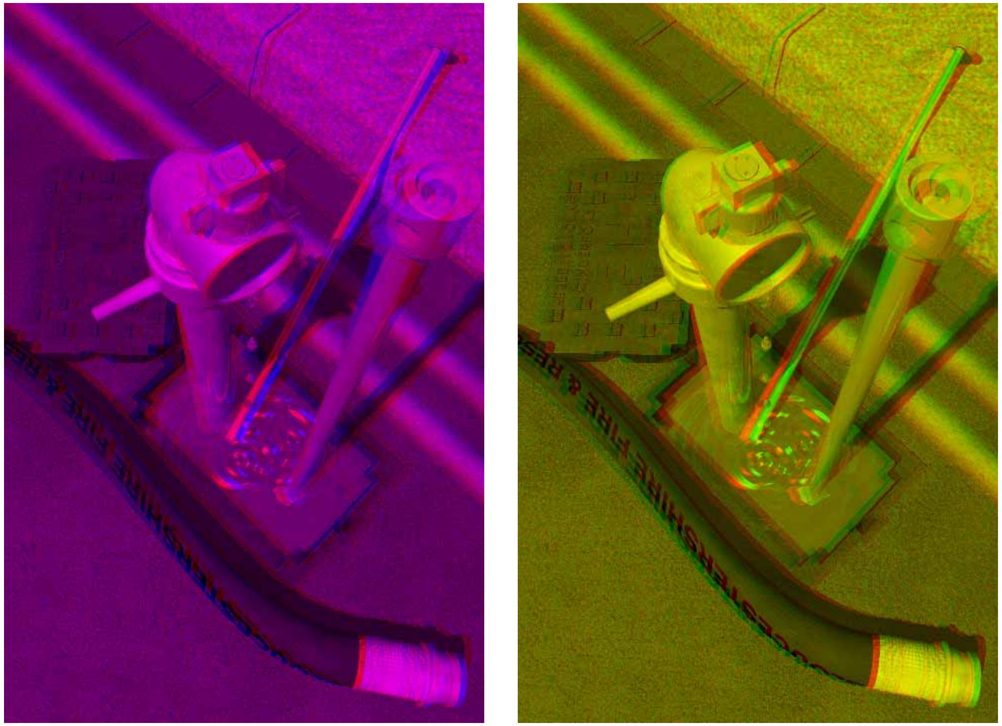
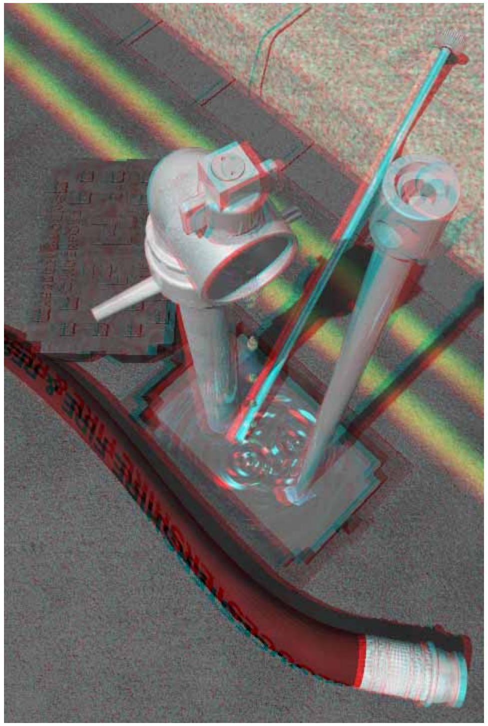


Figure 26: Using the Blue Only (left) or the Green Only (right) pure Anaglyphs can be generated. Whilst these only portray the scene in shades of grey, they are the easier to view.



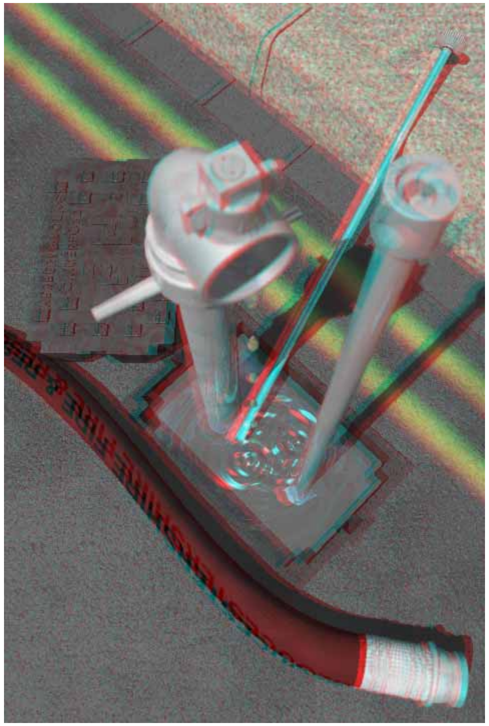


Figure 27: Whilst Depth of Field improves the illusion of reality in traditional renders it can make Anaglyph images more difficult and uncomfortable for the eye/brain to "fuse". Compare the Anaglyph with Depth of Field (right) to the one without (left).

Depth Controls

Both the Anaglyph and the Depth of Field (see page 7~51) features both want to use the camera's Y-axis length to control the "important" distance. With Depth of Field this distance equates to the plane in the scene where objects are perfectly in focus, whilst with Anaglyph it specifies the point in the scene that should appear on the "surface" of the monitor screen. All other distances are either (out of focus or) inside or outside the surface of the monitor screen.

The way that Anaglyph 3D stereo image generation is implemented as part of the **RENDER** or **QUICKRENDER** commands allows both options to function independently. (Depth of Field is controlled using the camera's actor bar in the *Action Dialog*.) The camera's Y-axis length can be set to determine the plane in the scene that is in perfect focus. By using the Distance Multiplier (see page 55) the position determined by the camera's Y-axis size, i.e. the portion of the scene that is required to appear on the "surface of the monitor", can be varied independently, i.e. without having to physically alter the camera's Y-axis size.

The Depth of Field feature certainly reduces that "computer generated" look in renders because we are used to seeing photographs and television images that are not in *perfect focus throughout the image. However, in real life the eye's accommodation reflex allows us to see everything in perfect focus. Similarly, in most circumstances, Anaglyph images should exhibit this perfect focus; introducing Depth of Field in an Anaglyph image will make the image uncomfortable and difficult to "fuse".*

⇔ **Distance Multiplier (1.000)** slider ~ The Distance Multiplier determines the plane in the scene that will appear on the "surface of the monitor screen" in the resulting Anaglyph image by controlling the camera's Y-axis size. Implementing this value as a multiplier allows the Cameras Y-axis length to be varied without physically altering the size of the camera. This is important because the camera's

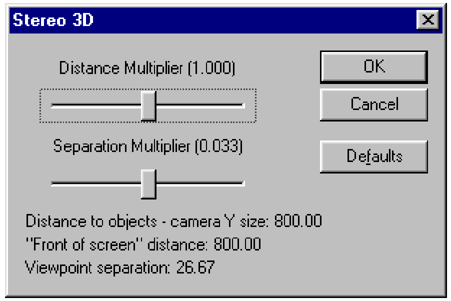


Figure 28: The Stereo 3D dialogue showing the values for the Anaglyph scene below.

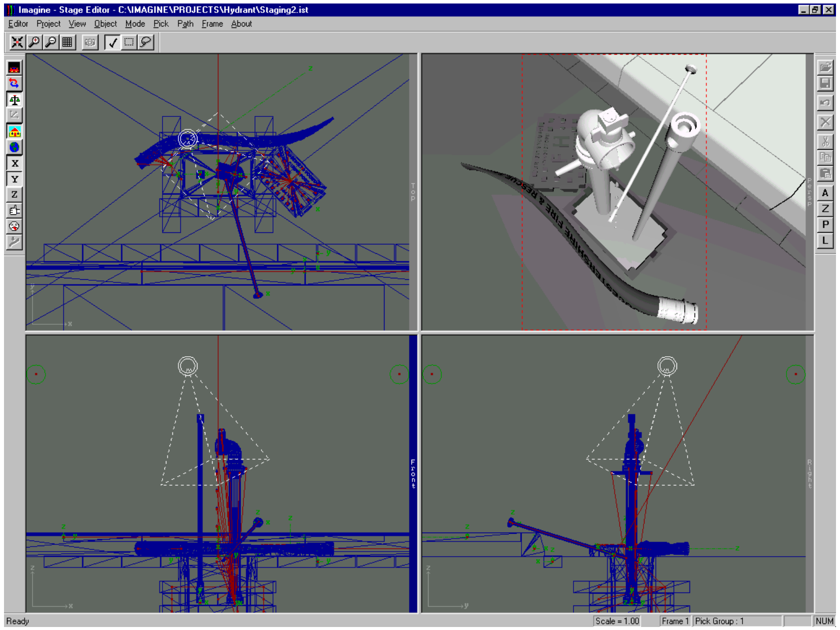
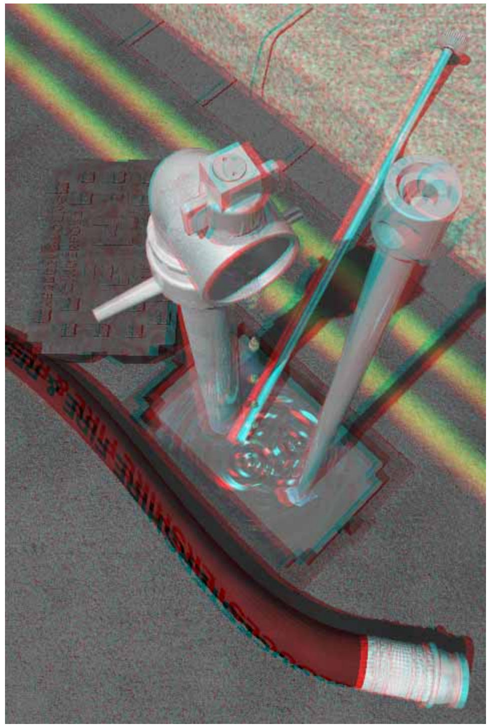
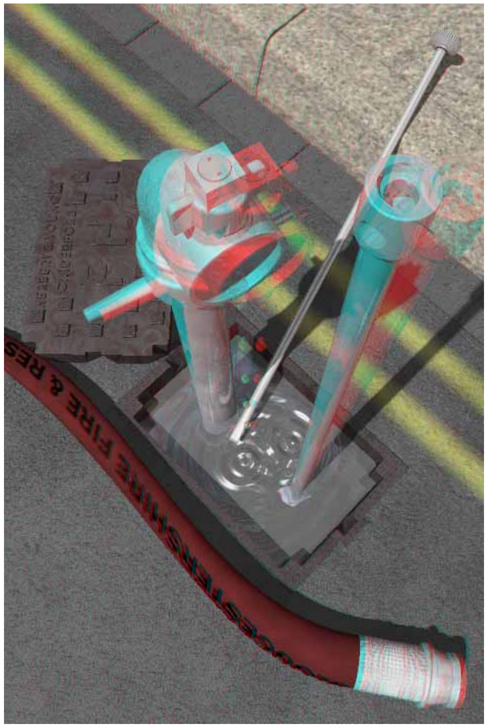


Figure 29: Using the CAMERA LINES option, where the "surface of the monitor" will lie in the scene can be accurately placed (above). In this example the camera's focal plane lies approximately halfway up the standpipe so that areas of the Anaglyph image (right) fall "inside" and "outside" of the monitor screen.



Y-axis size may have been previously set to a specific size to achieve a certain point of focus in the scene for the Depth of Field option.

By default, the slider is centralised, giving a multiplier of 1. This simply multiplies the camera's Y size by 1, which obviously equates to its actual size. Multiplying it by fractional values between 0 and 1 (dragging the slider to the left) results in a decrease in the camera's Y-axis size, in effect bringing the "monitor screen" closer to the camera. Increasing the Multiplier value up to a maximum of 10 (dragging the slider to the right) increases the size of the camera's Y-axis size, in effect moving the "monitor screen" away from the camera.



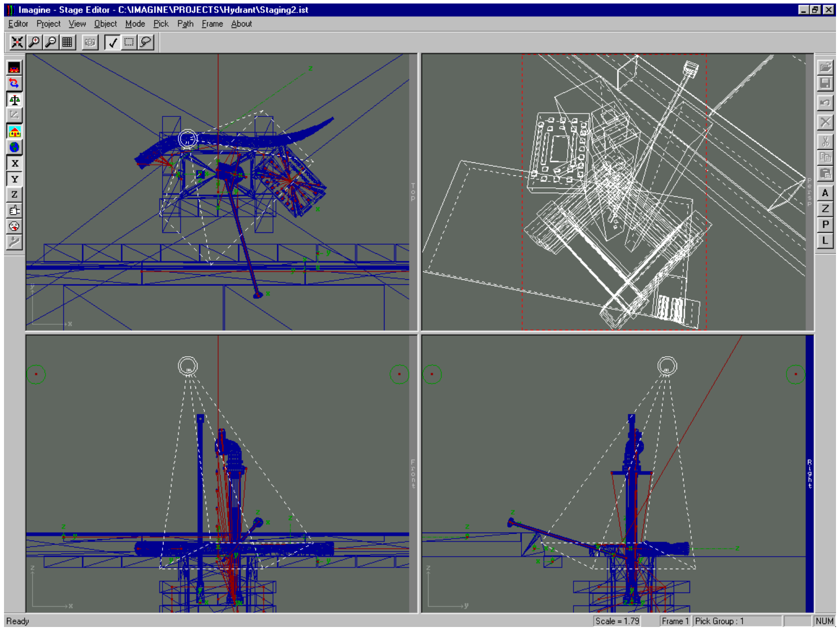
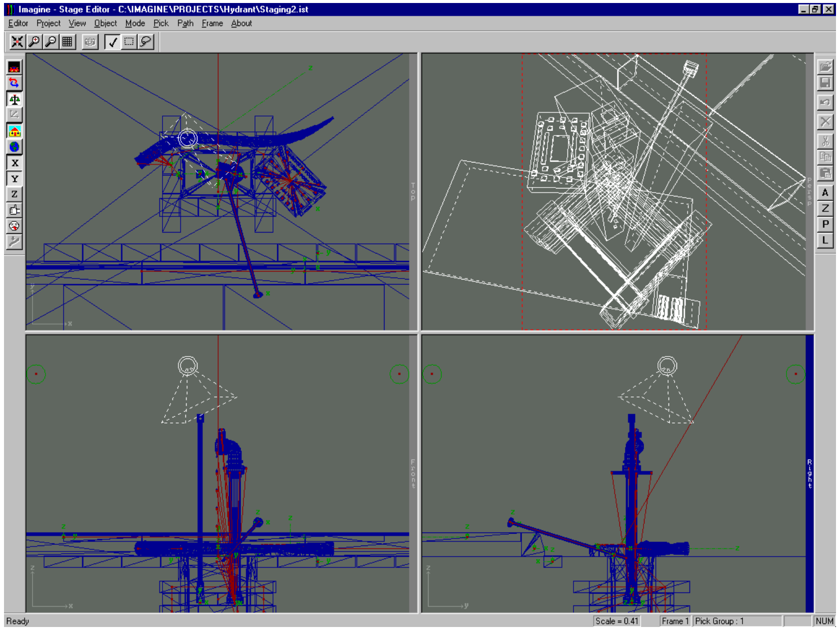


Figure 30: Using the CAMERA LINES option, where the "surface of the monitor" will lie in the scene can be accurately placed (above). In this example the camera's focal plane lies at road level so that all of the Anaglyph image (left) projects "out" of the monitor screen. Because no areas are "behind the monitor screen", it is not quite so effective.

How can the camera's Y-axis size and thus the position in the scene where the "surface of the monitor" will lie in the resulting Anaglyph 3D stereo image be calculated? If the Depth of Field feature is not being implemented the problem does not arise. Whilst in the Stage editor the camera's Y-axis can be numerically specified using the **TRANSFORM** command if the distance is known or, and more usually, the camera's Y size can be interactively set. To maintain the chosen camera perspective and zoom scale on the local X and Y axes, i.e. ensure that the **interactive LOCAL**, **X-axis** and **Y-axis** toolbar buttons are selected or press the keyboard shortcuts "***s***" and "***shift-z***". Then, with **CAMERA LINES** active (see



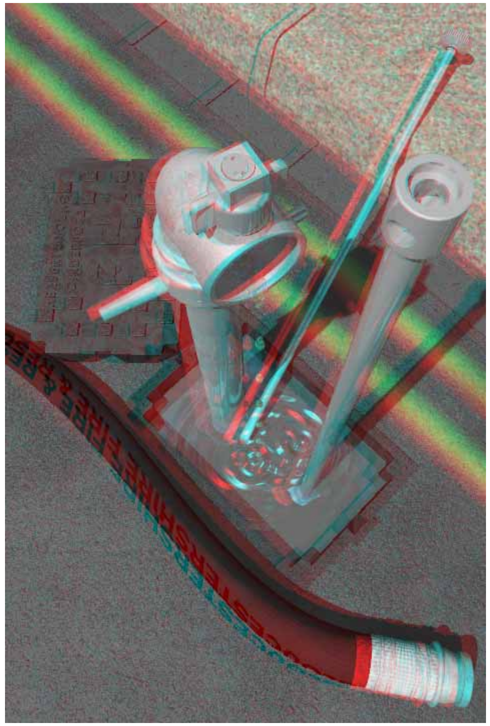


Figure 31: Using the CAMERA LINES option, where the "surface of the monitor" will lie in the scene can be accurately placed (above). In this example the camera's focal plane lies at the top of the scene so that all of the Anaglyph image (right) projects "in" to the monitor screen. Because no areas are "infront of the monitor screen", it is not quite so effective.

page 7~10 in the I Files) the “plane of focus” can be positioned in the scene corresponding to where the “screen of the monitor” is to appear in the resulting Anaglyph. In the *Detail editor* using the QUICKRENDER command the current VIEWING DISTANCE (see page 1~18) can be determined and the distance from the world’s origin (0,0,0) subtracted from it.

The Distance Multiplier in the *Stereo 3D dialog* should of course, be left at 1.000, i.e. when the camera’s Y-axis size is the actual distance into the scene where the “monitor screen” is to lie.

When a specific Y size has already been used to describe the point in the scene of perfect focus for the Depth of Field function, a slightly different approach should be used. In the *Stage editor* (Depth of Field using the camera’s Y-axis size is not available in the other *Editors*) ensure that the project has been saved reflecting the current Y-axis size of the camera. Interactively manipulate the camera’s X and Y axes together as detailed above with CAMERA LINES active to position the focal plane at the point in the scene that is to appear on the “surface of the monitor”. Accept the change and using the TRANSFORMATION command ascertain the camera’s Y-axis size from the size option. UNDO this operation to restore the original camera size settings.

Now it is simply a matter of dividing the camera’s original Y-axis size by the value gained by interactively manipulating the camera axes. This will result in the value for the Distance Multiplier. Alternatively, drag the Distance Multiplier slider to the left or the right until the value for the “Front of screen distance” (shown towards the bottom of the *Stereo 3D dialog*) matches the value derived from the interactive manipulation of the camera’s size.

The general rule is that decreasing the camera’s Y-axis size has the effect of bringing the “monitor Screen” closer to the camera and vice versa. The monitor screen is the viewport into the Anaglyph “volume” and as such any object in the scene “too large” to fit through the monitor should lie behind the screen of the monitor.

Towards the bottom of the *Dialog* the “Distance to objects - camera Y size: #” reports the camera’s Y-axis size as it has been specified in the scene being rendered. This value never changes — unless the camera's Y-axis size is altered in the *Stage editor* or the interactive PERSPECTIVE toolbar button is used in any of the *Editors*. It is there for reference only, when a "multiplier" needs to be calculated. The important value to note is the "Front of screen distance: #". Initially this will show the same distance as the "Distance to objects - camera Y size: #" because the Distance Multiplier defaults to 1. As the slider is dragged to the left the cameras Y-axis size will decrease, whilst dragging the slider to the right will increase the cameras Y-axis size. The "Distance to objects - camera Y size: #" will continuously update to reflect the changes the Distance Multiplier value is having on the cameras Y-axis size.

> ⇔ **Separation Multiplier (0.033)** slider ~ The Separation Multiplier relates to the horizontal offset between the two positions that the scene will be rendered from to fuse into a single stereo 3D image. Again, this value is expressed as a multiplier, which defaults to one thirtieth of the camera's Y-axis size. Combined with the Distance Multiplier (which in effect fixes a point in the scene that acts as a target for the camera to track between these two positions) these two depth controls determine the "volume" of the Anaglyph scene and the success with which it can be viewed.

This offset value, also known as the "Base" defaults to 0.033, or one thirtieth of the distance to the position in the scene that will be on the "surface of the monitor"; the cam-

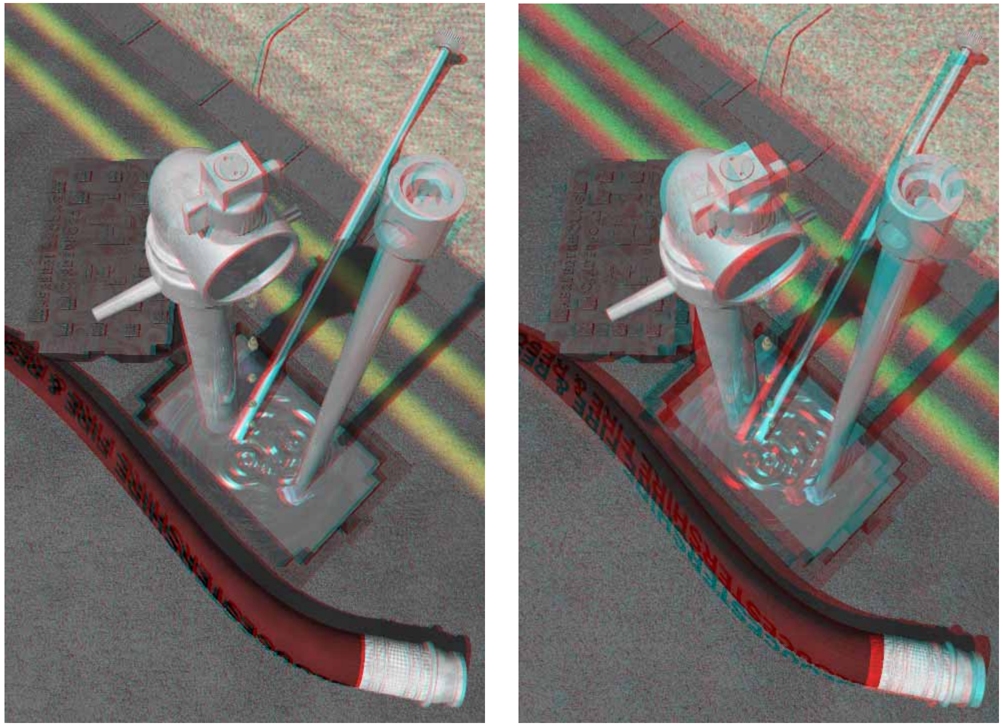


Figure 32: Varying the separation Multiplier can dramatically change the Apparent scale of the image. Hyper-stereo results if the distance is greater than the default, i.e. larger multiplier values (right) and hypo-stereo where the distance is less, i.e. smaller multiplier values (left).

era's Y-axis size. The "one thirtieth" value is derived from the statistical average horizontal offset between human eyes; the inter-ocular distance. This distance averages 65 mm and is calculated according to the average field of view. Standing in front of a window, it is impossible to see the horizon AND the window at the same time. However, retreat from the window to about two metres and human vision can accommodate both the horizon and the window surround. At a distance of two metres from the window, the average inter-ocular distance of 65mm corresponds to one thirtieth of this distance. This ratio gives the most comfortable viewing of Anaglyph stereo 3D images. When the base distance is greater than the inter-ocular distance the resulting view appear as though the viewer is a giant looking at a miniature scene; hyper-stereo. Where the base is less than the inter-ocular distance, hypo-stereo results; everything appears too large as though the viewer is a midget.

The viewpoint separation at the bottom of the *Dialog* is the related value, being dependent on the camera's Y-axis size, i.e. one thirtieth of it.

Imagine renders the scene with the camera directed at a specified target — that determined by the end of the camera's Y-axis — but from two slightly different viewpoints. The colour components from these two renders are then combined in such a way as to colour encode each image for the left and right eyes. The following parameters determine how the colour channels in each original image are mixed to produce a single Anaglyph 3D stereo image.

Colour mixing

Accepting that colour perception is notoriously subjective, it is recommended that both the Color Mixing and Intensity Balance controls be left at their default values. These default values have proven to gives the best results in most cases. So unless a specific result is required or you are competent in Anaglyph generation and colour mixing strategies, these two controls are best avoided.

However, for the more experienced user, the following discourse outlines the theory behind the two parameters and the complete control over the mixing process that they provide…

To produce a greyscale Anaglyph each of the two rendered views are "reduced" to greyscale representations. Then, the green/blue components from the left image are removed and replaced by the green/blue components from the right. This assumes the default settings when the red lens should cover the left eye and the green/blue the right. It is during this process that the green/blue channels end up identical, the entire image appearing as shades of grey.

A similar process is used when combining the two rendered views to produce a colour Anaglyph. The greyscale conversion is again used, but this time the left view is allowed to maintain some of it's red component and the right view some of it's green/blue component; again assuming the default lens arrangement.

The obvious question is why not used the entire unaltered red component in the left image and the unaltered green/blue components in the right. The reason is that problems arise when there are bright and saturated areas of colour in the renders. A bright red object appears correct to the left eye but the green/blue components of this object will appear to the right eye as "dark". This obviously confuses the brain as the left eye is seeing a bright object whilst the right eye observes a dark one, thus making the Anaglyph image difficult to view.

To overcome this problem the red component needs to be "dimmed" a little for the left eye and the green/blue components "brightened" for the right. The way to achieve this is to mix a greyscale version with each of the colour channels in each render before composing the final colour Anaglyph image. This process makes the Anaglyph far easier to view whilst still maintaining much of the colour information.

Having briefly discussed the process, the following controls should make more sense…

- **Color Mixing** button ~ Selecting the Color Mixing button will display a dialog appropriate to whether the Color or Greyscale Anaglyph type has been selected.

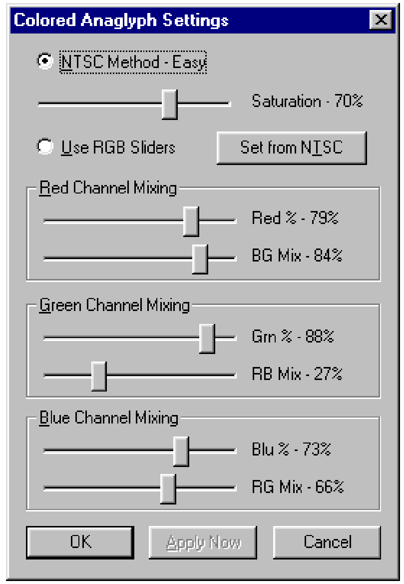


Figure 33: The Colored Anaglyph Settings dialogue showing the default values.

The *Colored Anaglyph Settings dialog*, displayed by selecting the Color Settings button on the *Anaglyph Controls dialog* offers the following parameters governing the way colour Anaglyphs are generated…

- **NTSC Method - Easy** radio button ~ The NTSC Method uses a standard mixing ratio to convert the original colour image into a greyscale interpretation which is subsequently mixed with the individual colour channels prior to conversion to the Anaglyph image.

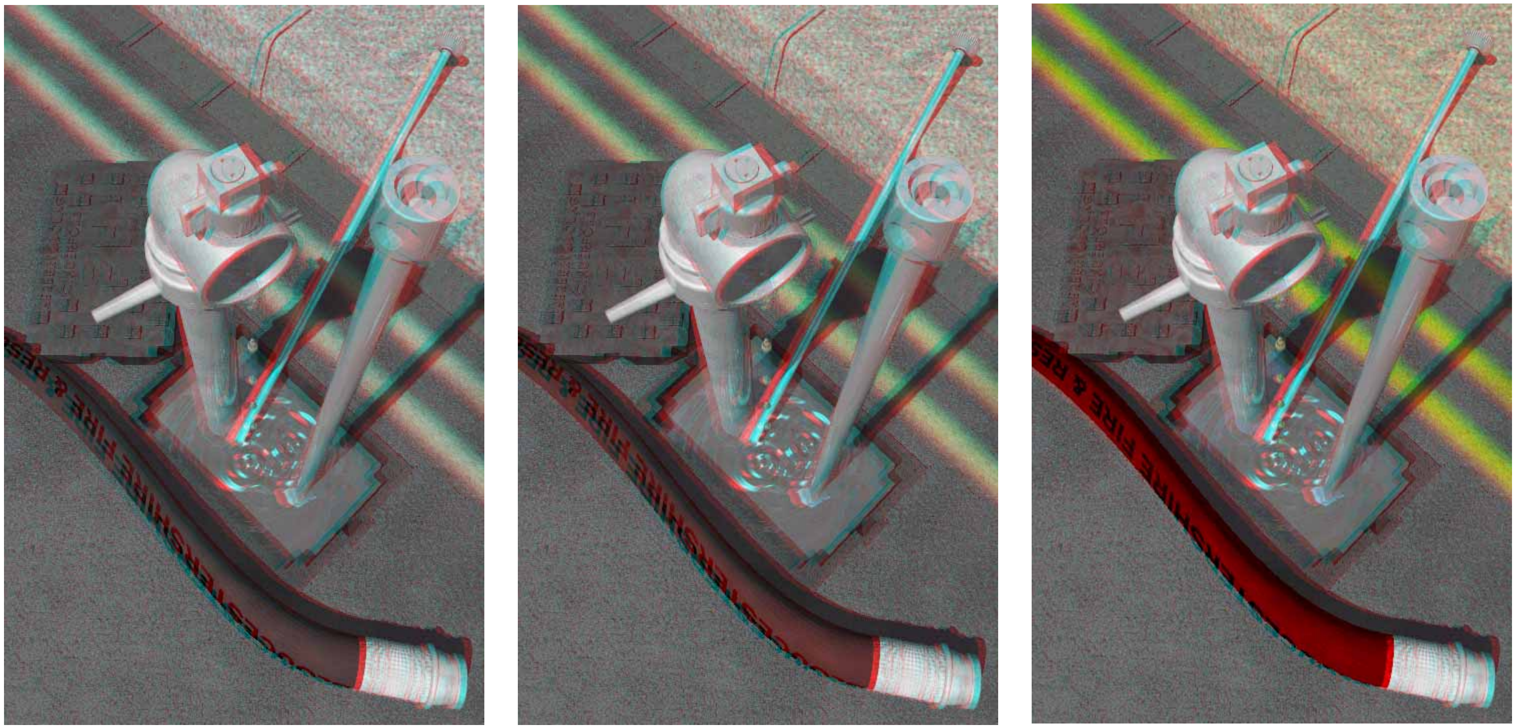


Figure 34: The results of varying the Saturation parameter, 25% (left) 50% (middle) and 100% (right). Notice how as the Anaglyph is composed using more colour, it becomes slightly more difficult to "fuse" the 3D stereo image.

⇔ **Saturation - (%)** slider ~ The Saturation control determines how much of the original colour components are maintained during the conversion process. A larger value results in more of the original colouring being preserved and vice versa. At default value of 70% there is a 70/30 mix of the original version with the greyscale interpretation of it before the colour components are separated.

This is probably the only parameter that should be varied and even then, only to adjust scenes that have lots of bright, fully saturated colours. When experimenting with this control ideally it needs to be kept as high as possible without sacrificing the ability to easily view the Anaglyph.

The NTSC Method in conjunction with the Saturation slider gives a “one slider” control over the variables.

⊙ **Use RGB Sliders** radio button ~ Selecting this option gives full and detailed control over the colour mixing; the initial defaults relate to the standard NTSC mixing value. Using this level of control requires a detailed knowledge of colour mixing strategies and requires a real need for fine-tuning the resulting Anaglyph 3D stereo image.

❑ **Set from NTSC** button ~ Selecting this option will initialise each of the RGB sliders based on the Saturation slider setting and the corresponding standard NTSC ratios for obtaining greyscale conversions.

The colour channel mixing values will depend on the saturation value chosen.

Red Channel Mixing:

⇔ **Red (%)** slider ~ Specifies the percentage of the original red component that is to be maintained during the mixing process; the default value being 79%.

⇔ **BG Mix (%)** slider ~ The remaining 21% of the red channel is a mixture of blue and green; the default being 84% blue and 16% green.

Thus, for a saturation value of 70% of the original colour, 79% of the red component is mixed with 84% blue and 16% green to give the greyscale version with the red “enhancement”.

Green Channel Mixing:

⇔ **Grn (%)** slider ~ Specifies the percentage of the original green component that is to be maintained during the mixing process; the default value being 88%.

⇔ **RB Mix (%)** slider ~ The remaining 12% of the green channel is a mixture of red and blue; the default being 27% red and 73% blue.

Thus, for a saturation value of 70% of the original colour, 88% of the green component is mixed with 27% red and 73% green to give the greyscale version with the green "enhancement".

Blue Channel Mixing:

⇔ **Blu (%)** slider ~ Specifies the percentage of the original blue component that is to be maintained during the mixing process; the default value being 73%.

⇔ **RG Mix** slider ~ The remaining 27% of the blue channel is a mixture of red and green; the default being 66% red and 34% green.

Thus, for a saturation value of 70% of the original colour, 73% of the blue component is mixed with 66% blue and 34% green to give the greyscale version with the red "enhancement".

For those that have their own mixing strategies, the following formula indicates a 70/30 mix of the original red component, with a standard NTSC greyscale conversion ratio:

$$\textbf{New Red} = 0.70 \times R + 0.30 \times (0.30 \times R + 0.59 \times G + 0.11 \times B)$$

The Greyscale *Anaglyph Settings dialog*, displayed by selecting the Greyscale Settings button on the *Anaglyph Controls dialog* offers the following parameters governing the way a greyscale Anaglyph is generated…

⦿ **NTSC B/W Mixing Ratios** radio button ~ The NTSC B/W Mixing Ratios uses a standard mixing ratio to convert the original colour image into a greyscale representation.

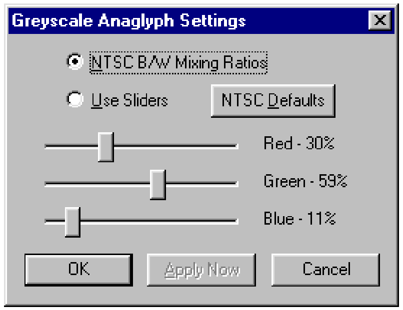


Figure 35: The Greyscale Anaglyph Settings dialogue showing the default values.

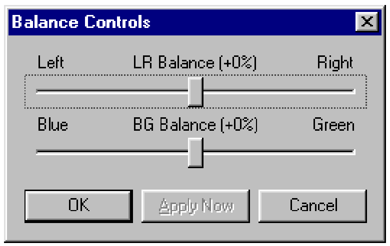


Figure 36: The Balance Controls dialogue showing the default values.

This standard ratio was arrived at following the analysis of a large number of test subjects and their reactions to the conversion of colour images to greyscale when viewed on a black and white television.

- **Use Sliders** radiobutton ~ Selecting this option overrides the standard NTSC ratios allowing complete control over the individual colour channels and their conversion to a greyscale representation.

- **NTSC Defaults** button ~ Selecting the NTSC Defaults option resets the individual colour channels to the standard NTSC greyscale conversion ratio.

- **Red (%)** slider ~ Specifies the "weighting" given to the original red component that should be used in the greyscale conversion.

- **Green (%)** slider ~ Specifies the "weighting" given to the original green component that should be used in the greyscale conversion.

- **Blue (%)** slider ~ Specifies the "weighting" given to the original blue component that should be used in the greyscale conversion.

Intensity balance

The Intensity Balance controls allow for the final overall balance of the colour channel

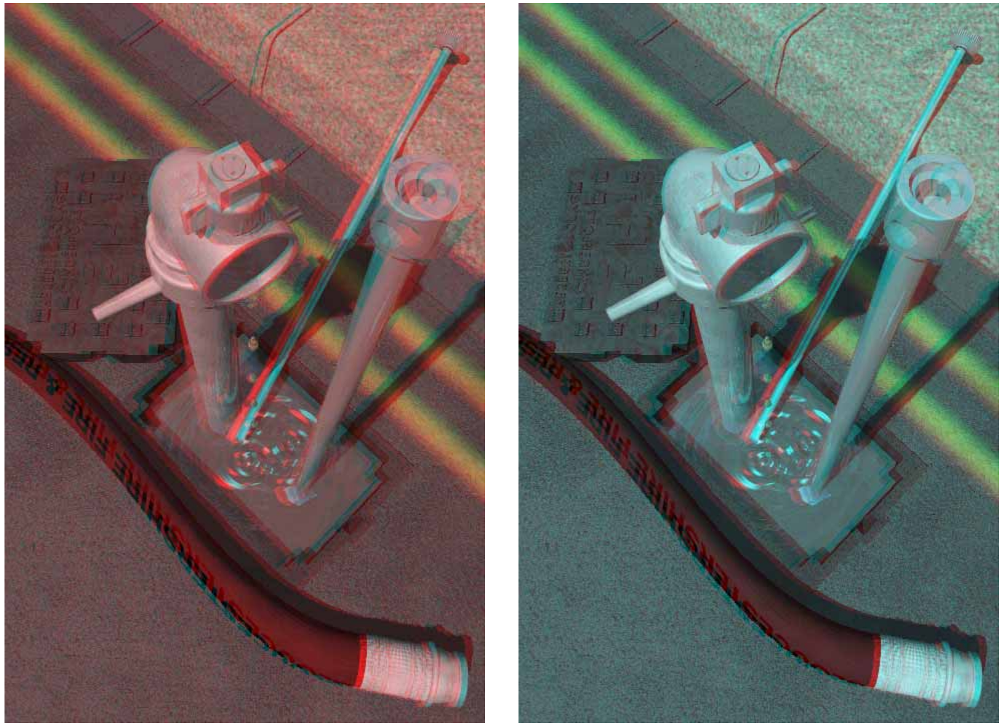


Figure 37: By varying the LR Balance the variations in Anaglyph lenses can be accomodated. Moving the slider to the left decreases the blue/green channel where the red lens appears darker (left) whilst moving the slider to the right reduces the red channel where the blue/green lens appears darker (right).

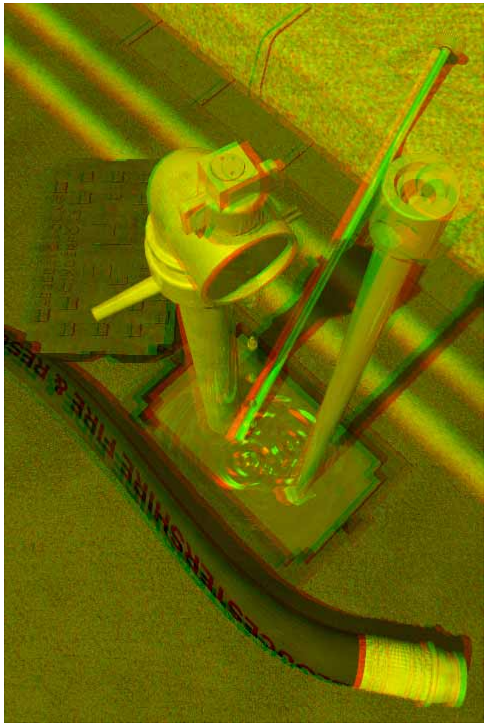


Figure 38: By varying the BG Balance the variations in the Anaglyph glasses blue/green lens can be accomodated. Moving the slider to the left decreases the blue component (left) whilst moving it to the right reduces the green component (right).

components to be tuned to the specific pair of Anaglyph glasses being used. For example, if the blue/green lens appears darker, the red component can be scaled down to compensate in order to balance the image being seen in the blue/green channel.

The BG Balance slider operates in a similar manner allowing fine-tuning to the "colour" of the blue/green lens. For example, if the lens appears to allow more blue light through than green (it appears bluer) adjusting the slider in favour of the green component causes the blue component to be reduced. However, once this slider is altered, the LR Balance is likely to be upset requiring that to be adjusted to match.

Intensity Balance button:

⇔**LR Balance (%)** slider ~ This balance control compensates for Anaglyph glasses that appear to have one lens producing a lighter or darker view than the other. Moving the slider to the right diminishes the intensity of the left component of the image whilst moving it to the left diminishes the intensity of the right component.

⬄ **BG Balance (%)** slider ~ This balance control compensates for Anaglyph glasses that favour either blue or green light through the blue/green lens. Moving the slider towards green reduces the amount of blue component in the Anaglyph image and moving it towards the blue reduces the green component.

❑ **Restore Defaults** button ~ The Restore Defaults button, not surprisingly, resets all the options and parameters to their default condition. Generally, these values can be left unaltered for the generation of acceptable Anaglyph images.

Interactive mixing

Aside from the depth controls, the type of Anaglyph, the lens configuration, the colour mixing and intensity balance are all post-processing operations. This allows the opportunity to vary these parameters before finalising the Anaglyph 3D stereo image. The following option does just that…

☑ **Render image, show this again** check box ~ Aside from the depth controls that govern the target and displacement of the camera the remaining treatment for encoding the Anaglyph image is all achieved through post processing. This enables the various colour and type controls to be manipulated interactively, by displaying the *Anaglyph dialog* prior to the image being finalised.

The *Anaglyph dialog* allows the type, eye colour preferences, colour mixing and intensity balance to all be varied before the composite image is finalised. This allows the Anaglyph 3D stereo image to be fine-tuned. Bear in mind though that the distance and separation parameters have already been calculated and cannot be varied.

Because the image has not been finalised, it cannot be saved until the current values in the *Dialog* have been accepted.

Figure 39: The Anaglyph dialogue displayed after both views have been rendered prior to being mixed for the Anaglyph image.

Ini file

The following is a list of the entries in the Anaglyph section in the "Imagine.ini" file found in the Windows folder and shows the default values. Reference to this file indicates which of the preceding parameters are stored, i.e. these are the values that will be used in subsequent ***Imagine*** settings…

[Anaglyph]

Type=0

    0= Anaglyph rendering is disabled

    1= Greyscale Anaglyph

    2= Colour Anaglyph

OtherView=2

    0= Blue only

    1= Green only

    2= Mix

RedView=1

    0= Right

    1= Left

LRBal=50

    Initial percentage slider setting in the *Balance Controls dialog*

BGBal=50

    Initial percentage slider setting in the *Balance Controls dialog*

NTSCGreyScale=0

0= NTSC B/W Mixing Ratios in the *Greyscale Anaglyph Settings dialog*

1= Use Sliders in the *Greyscale Anaglyph Settings dialog*

GSRed=30

Initial percentage slider setting in the *Greyscale Anaglyph Settings dialog*

GSGreen=59

Initial percentage slider setting in the *Greyscale Anaglyph Settings dialog*

GSBlue=11

Initial percentage slider setting in the *Greyscale Anaglyph Settings dialog*

NTSCColored=1

0= Use RGB Sliders in the *Colored Anaglyph Settings dialog*

1= NTSC Method - Easy in the *Colored Anaglyph Settings dialog*

NTSCSat=70

Initial percentage slider setting in the *Colored Anaglyph Settings dialog*

RedVal=79

Initial percentage slider setting in the *Colored Anaglyph Settings dialog*

RedBGBal=84

Initial percentage slider setting in the *Colored Anaglyph Settings dialog*

GreenVal=88

Initial percentage slider setting in the *Colored Anaglyph Settings dialog*

GreenRBBal=27

Initial percentage slider setting in the *Colored Anaglyph Settings dialog*

BlueVal=73

Initial percentage slider setting in the *Colored Anaglyph Settings dialog*

BlueRGBal=66

Initial percentage slider setting in the *Colored Anaglyph Settings dialog*

Distance=50

Initial percentage slider setting in the *Balance Controls dialog*

Separation=50

Initial percentage slider setting in the *Balance Controls dialog*

refined Bones and Kinematics

Before considering the improvements, let us briefly refresh our memories concerning the complex subjects of Bones, Kinematics and States. For those who have the I Files, please refer to Chapter 6.

Bones

In ***Imagine***, Bones are plain axes hierarchically grouped that control the polygonal geometry to which they are associated; the object geometry must form a contiguous mesh. A "boned" object must be composed of at least two axes (one to anchor the object's faces and one or more to control the objects deformation) and these axes must be hierarchically grouped in the correct order. Each "bone axis" must be a plain axis and have both Small Bone Subgroup and a Big Bone Subgroups of faces assigned to it. The Small Bone Subgroup are those faces that move and rotate with the axis whilst the Big Bone Subgroup contains not only the Small Bone Subgroup faces but also those faces that will stretch or compress as the axis is manipulated. The number of faces defined for the Small Bone Subgroup should never exceed the number defined for the Big Bone Subgroup.

States

To use Bones involves an understanding of how they interact with States; a "Default" State must be defined. States allow various "instances" or key poses of an object to be saved within the object's definition. This includes the object's surface appearance, position, size, alignment grouping and shape.

Only the position of "child" objects can be stored within the State data (the position of the "parent" object is stored within the "staging file") – hence the need for at least two axes. The "parent" axis contains all the State information, although further State information for "child" objects is acceptable. The first axis "under" the "parent" axis for the object must also have all the faces comprising the object defined for both its Small and Big Bone

Subgroups. This is then stored in the Shape State data type as a reference of how the object first appears prior to being deformed. The "Default" State should have at least the Grouping and Shape data types selected. This information provides the "basic" shape of the object and the relationship of the "child" axes to the "parent". From this State, any number of further States can be created varying any of the parameters of the object. ***Imagine*** can then smoothly interpolate between them and the "Default" State. Further States should not have the Shape data type selected unless the actual geometry has been manually altered in addition to the deformation created by the manipulation of the Bones.

Kinematics

Kinematics is the method of manipulating a correctly grouped hierarchy of objects in a natural manner. It comes in two flavours: Forward and Inverse Kinematics.

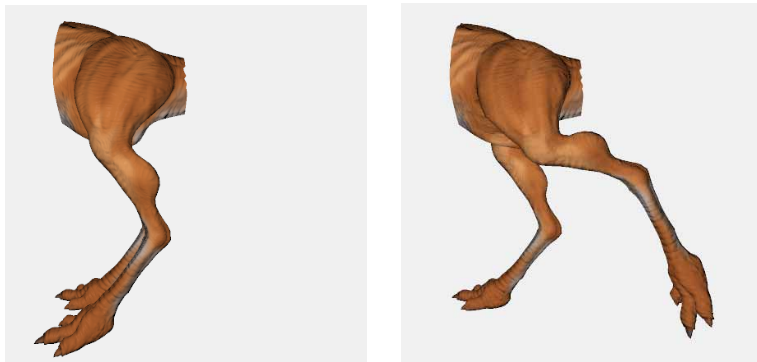
Forward kinematics is the traditional method of manipulating groups of objects where an object or axis is picked and all those "below" it in the grouping chain become picked provided the Pick GROUPS editing mode is active. Manipulating the picked object/axis results in all those in the chain below following their "parent", maintaining their relationship to the object/axis immediately above them. In other words, manipulations "move" down the grouping chain with the individual object/axes below the "parent" maintaining their relationships to one another.

Inverse kinematics on the other hand "move" up the chain. A "child" object/axis is picked and all those above it in the grouping chain become picked. Then, when the "child" is manipulated all those above it react according to their individual constraints. For this to occur the CONSTRAIN toggle command should be selected, constraints should be applied using the FREEZE command and the Pick OBJECT mode must be active.

Generally, Inverse kinematics are used to create the overall pose because of the "natural" way it controls the grouping chain. Forward kinematics can then be used to fine-tune the position and or alignment of each object/axis.

Constraints

As was just mentioned, for Inverse kinematics to function effectively the axes within a grouping hierarchy should be constrained using the **FREEZE** command. This limits their movement and/or rotation allowing the axes to mimic real-world joints.



Changes

Having covered the basics again, what has been changed?

Changes have been made to the inverse kinematics feature making it faster to use with smoother feedback allowing poses to be set more intuitively.

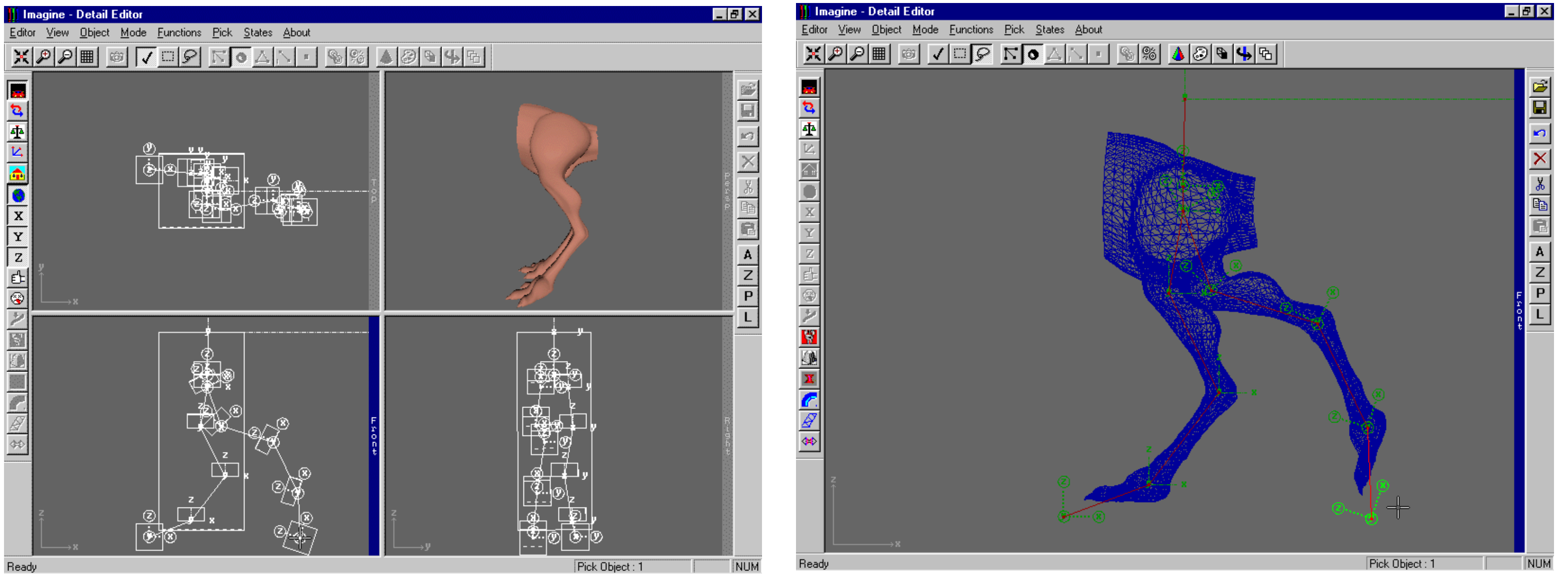
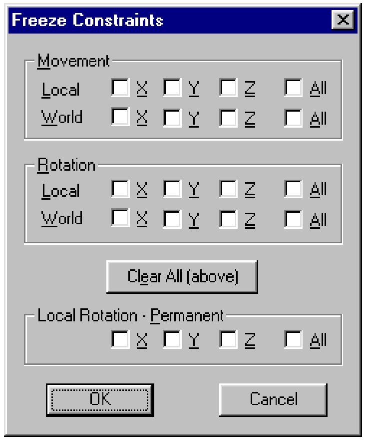


Figure 40: Inverse kinematics are now smoother and faster, not exhibiting the "jerky" movements sometimes encountered before (left). This leads to more naturally posed joint articulations (top right). Bones axes that have had constraints applied to them are now clearly indicated (bottom right).



Bones have now been improved with a pop-up *Dialog* displayed whenever a Bones axis is picked — with the right mouse button — to constrain it's movement and/or rotation. This saves a good deal of time when constraining many axes on complicated "boned" objects. Additionally, a Bones axis that has had constraints applied to it now displays a marker clearly identifying it as a constrained Bones axis.

Finally, local rotational constraints applied to a Bones axis can now be saved within the object file. This obviously, can save a good deal of work when it comes to posing objects in subsequent work sessions. You will notice from the *Freeze Constraints dialog* (shown) that the Local Rotation - Permanent area is towards the bottom. These are the options that are saved with the object. This is useful where joints are of the hinge variety and can only ever rotate in the one axis. The temporary constraints above, both movement and rotation operate as before; the Clear All (above) button is a one click option to remove any of the temporary constraints. This in effect, is the same as the **RELEASE** command but enables constraints to be removed without quitting the *Dialog*. It should be noted that neither the **RELEASE** or **RELEASE ALL** commands affect any permanent rotational constraints.

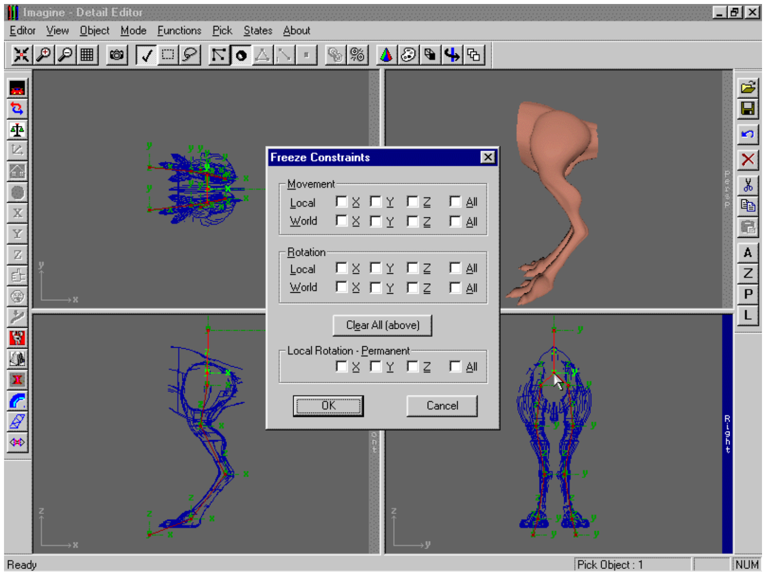


Figure 41: The Freeze Constraints dialog displayed by right-clicking over a Bones axis.

Whilst the existing DXF import/export filter does allow objects to be exchanged with other programs, it can be a cumbersome and unwieldy format. The DXF file format produces file sizes that are typically four to five times the size of proprietary three-dimensional object formats (being a “plain text” file). This situation has been improved with the inclusion of additional…

Import/Export filters

Two new import and export filters have been introduced during the Constant Upgrade Programme for two of the more popular “other” three-dimensional modelling and animations programs, namely *3D Studio* and *Lightwave*.

Import

> 3DS ~ ***Object-Import menu*** ~ Selecting the **Import-3DS** command displays the standard operating system “Open” file requester allowing the appropriate 3D Studio object to be located and imported. The file type suffix “.3DS” is automatically selected causing only those files with this suffix to be displayed.

Objects that were grouped together within *3D Studio* are imported into ***Imagine*** as separate objects. All the objects are imported but grouping assignments are ignored producing a number of separate objects (which can of course, be re-grouped within ***Imagine***).

> LWO ~ ***Object-Import menu*** ~ Selecting the **Import-LWO** command displays the standard operating system “Open” file requester allowing the appropriate *Lightwave* object to be located and imported. The file type suffix “.LWO” is automatically selected causing only those files with this suffix to be displayed.

Objects that were grouped together within *Lightwave* are imported into ***Imagine*** as a single object.

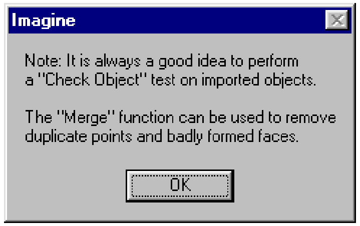


Figure 42: When importing "foreign" objects into Imagine, it is always a good idea to use the Check Object command. If any problems are encountered, the Merge command will usually rectify them. The displayed dialog serves as a reminder,

Whenever "foreign" objects are imported for use in ***Imagine*** (and this includes those in the DXF format) the **CHECK OBJECT** command should be used (see page 2~62 in the I Files). If this reports any problems with the imported object the **MERGE** function (see page 2~70 in the I Files) should be used. This is important because other programs treat polygonal geometry differently; selecting **MERGE** will remove duplicate points, edges, faces and generally optimise the object to suit ***Imagine***. A *Dialog* is displayed whenever an object is imported using the above two commands to reinforce this important step.

Due to the variety of ways that three-dimensional modelling program handle the definition of surface attributes, objects imported using these two filters are provided with only the Base Color defined as white and the Phong Shaded attribute set. Furthermore, all edges are treated as "smooth"; any edges that were defined as "sharp" in the originating program will have to be redefined once imported into ***Imagine*** using the **MAKE SHARP EDGES** command (see page 2~50 in the I Files).

Export

3DS ~ ***Object-Export menu*** ~ Selecting the **Export-3DS** command displays the standard operating system "Save As" file requester allowing the object to be named and a storage location found.

The file type suffix ".3DS" is automatically appended to the name specified and the polygonal object exported in a format that *3D Studio* will recognise.

LWO ~ ***Object-Export menu*** ~ Selecting the **Export-LWO** command displays the standard operating system "Save As" file requester allowing the object to be named and a storage location found.

The file type suffix ".LWO" is automatically appended to the name specified and the polygonal object exported in a format that *Lightwave* will recognise.

Only point based objects can be exported for use in *3D Studio* and *Lightwave*; a warning *Dialog* will be displayed if a plain axis or spline

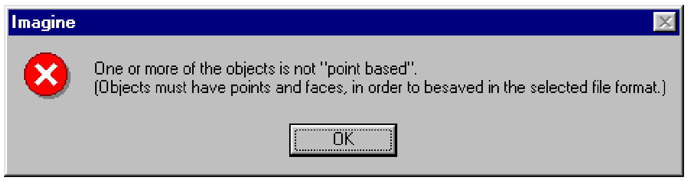


Figure 43: A Warning dialog is displayed when exporting objects if one or more of them are not point based.

object has been picked allowing the opportunity for it to be de-selected. Also, because of the difference in the ways the programs handle surface attributes only the basic attributes are defined in the exported files.

Figure 44: A Warning dialog is displayed if grouped objects are selected for export. This is due to the differences in the way each program handles groups of objects.

Additionally, grouped objects cannot be exported. Either **UNGROUP** the objects and save them to individual files, or enter **Pick OBJECT mode**, pick each object in turn and export it. Multiple objects can be exported with the file requester being displayed equal to the number of objects picked; each object is exported in the order that it was picked.

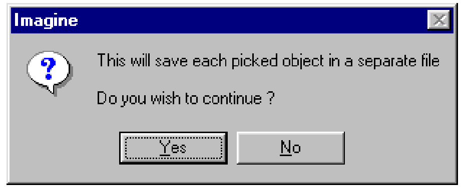


Figure 45: Multiple objects can be exported; the file requester being diplayed equal to the number of objects picked. Each will be saved to a separate file.

Deciding on the complexity of the object early on is not always possible, especially if the object already exists or is acquired from another source. Consequently, there was a need for a tool that allowed the optimisation of object geometry. The following command does precisely this; the geometric complexity of an object to be simplified and the results viewed whilst this is happening…

Reduce Points

Building complex geometric objects with the tools available is, with a little practise, fairly straightforward. With a fast processor and plenty of memory, it is not usually necessary to worry about the "efficiency" of the object too much when rendering a single image. However, as the scenes being built become ever more complex or the possibilities of animation are explored — leading to longer render times — more thought should be given to keeping objects geometrically "simple". The use of the minimum number of points, edges and faces to convey the shape of the object, without the polygonal nature of its construction becoming obvious, becomes ever more important. The **REDUCE POINTS** tool allows this optimisation to take place during or at the end of the construction of an object.

Reduce Points ~ *Functions menu* ~ The **REDUCE POINTS** command is accessible whenever an object group or a single object is picked. Whilst the **Pick GROUP** mode is active only the "parent" object in the grouping hierarchy will be modified. Additionally, it also becomes available in the **Pick POINTS**, **Pick EDGES** and **Pick FACES** modes, provided more than one item (in whichever mode) has been picked. This is particularly useful for optimising certain areas of an object's surface. For example, flat areas on an object require few polygons to define them whilst curved surfaces require more. Picking just those areas and leaving areas of smoothly changing curves unselected, the point reduction need only be applied to the flat surface which requires very little geometry.

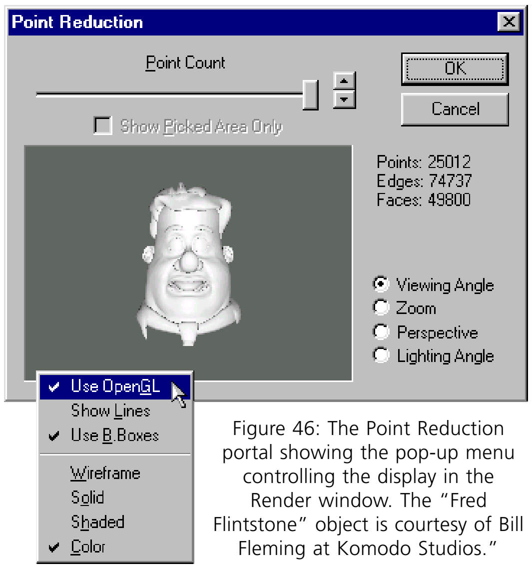


Figure 46: The Point Reduction portal showing the pop-up menu controlling the display in the Render window. The "Fred Flintstone" object is courtesy of Bill Fleming at Komodo Studios."

For those familiar with ***Organica*** and its point reduction controls it should be stressed that they work in totally different ways. ***Organica***, whilst allowing a more “intelligent” point reduction (flat surfaces are optimised to a greater extent than curved surfaces) is not dealing with polygonal object internally; these are generated when the object is exported. However, ***Imagine*** deals with polygonal objects from the start and this method of point reduction is not feasible.

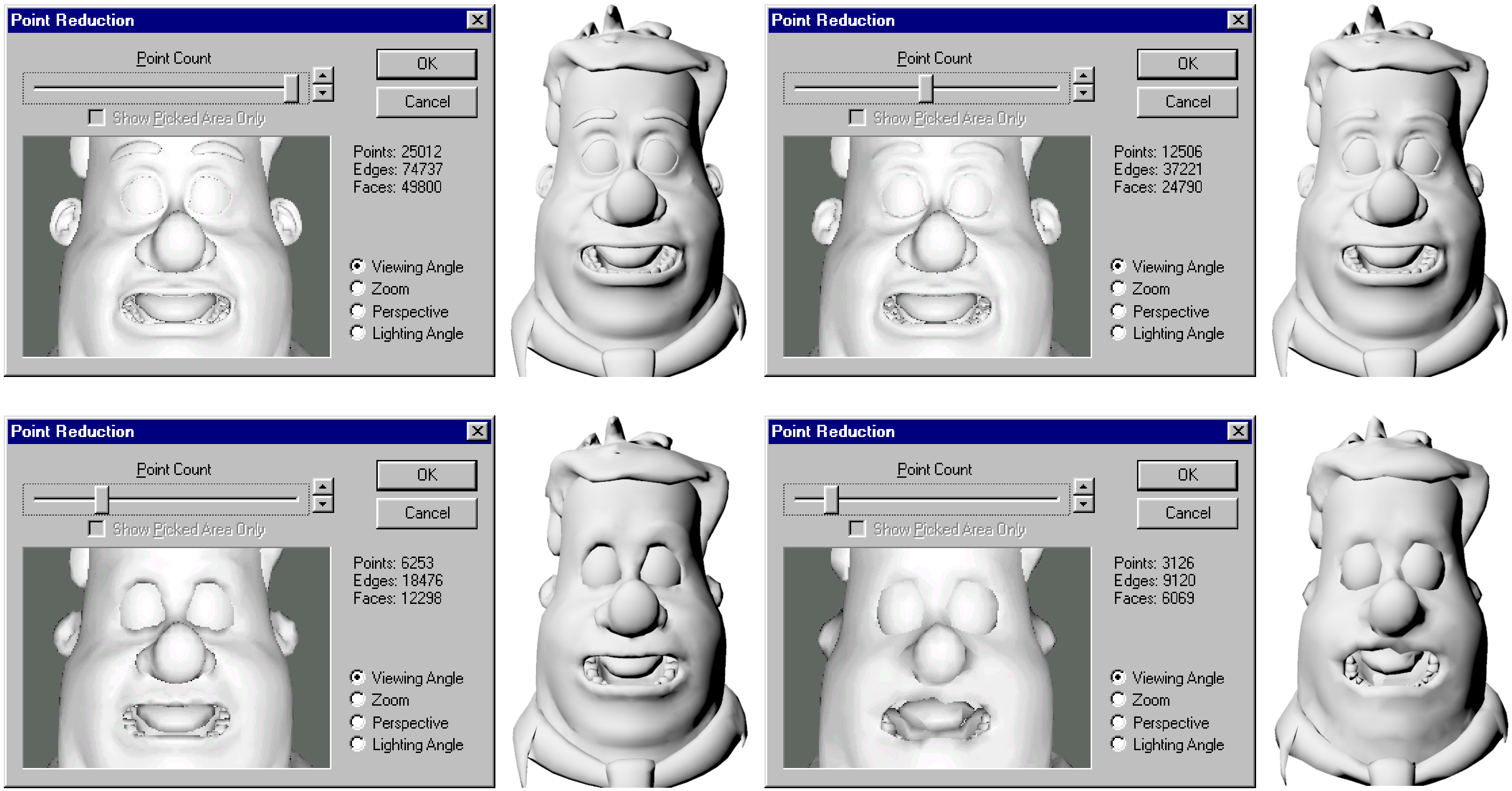
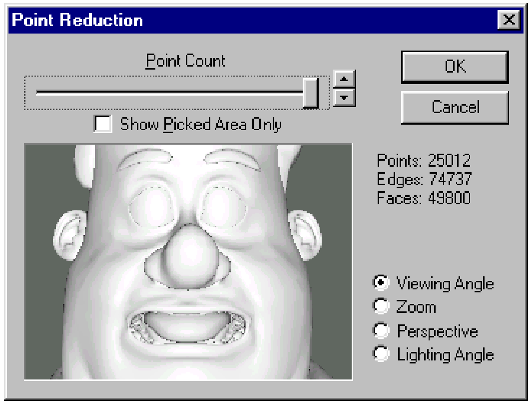


Figure 47: The Point Reduction portal showing the results of optimising the “Fred Flintstone” object: original complexity (top left) half complexity (top right) quarter complexity (bottom left) and an eighth complexity (bottom right). Reducing the overall complexity first enables areas that need to be selectively reduced to be identified.

REDUCE POINTS works across the whole object and does in fact optimise the fine detail areas first. Consequently, it is recommended that whenever the results in **Pick GROUP** or **Pick OBJECT** modes are not satisfactory, localised optimisation should be used.



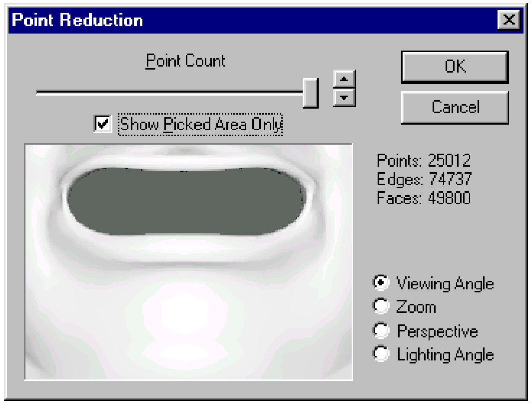


Figure 48: The Point Reduction portal showing the Show Picked Area Only option active (top). When selected the view is zoomed automatically to encompass the picked area (bottom).

Selecting the **REDUCE POINTS** command displays the *Point Reduction portal*, giving near instantaneous feedback (depending on the complexity of the object and the speed of the computer) of the results that simplifying the geometry has on the object's shape. The following parameters are all available from the *Portal*…

⇔ **Point Count** slider ~ Dragging the Point Count slider to the left (or using the "spinner" controls) reduces the object's geometric complexity. Moving the slider halfway to the left will reduce the number of points used to define the object by fifty percent.

When the *Portal* is first displayed, the slider is positioned fully to the right indicating no reduction; the point, edge and face values reflect the initial complexity of the object. Dragging the slider to the left, as its name suggests, will reduce the number of points used to define the object. ***Imagine*** will then calculate the number of edges and faces required to re-establish the object's coarser mesh.

The object's complexity will determine the length of time required to calculate the reduction in geometric detail.

Not only is there visual feedback from the display in the Render window indicating the appearance of the object as it's point count is reduced, but also to the right of the slider is an accurate count of the number of points, edges and faces comprising the object.

☑ **Show Picked Area Only** check box ~ This option allows certain areas of an object's surface to be viewed independently from the remainder of the object whilst reducing the point, edge and face count of that area. This option is particularly useful when a specific area of the object has been picked for optimising and viewing the remainder of the object would confuse the issue.

The availability of this option will depend on the editing mode currently employed in the *Editor*. Obviously, whilst in Pick GROUPS or OBJECT modes, specific areas of an object cannot be picked; consequently the option is unavailable. However, when multiple points, edges or faces are picked in their respective editing modes, this option becomes available for selection.

⦿ **Viewing Angle** radio button ~ The Viewing Angle option is the equivalent of the interactive ANGLE toolbar button (see page 1~18 in the I Files) in that it controls the orientation of the object in the Render window. Depressing the left mouse button whilst the cursor is over the Render window will cause the object to be redrawn according to the currently active display option (see below) thus allowing real-time feedback as the view is manipulated. Dragging the cursor from left to right rotates the view around the World's Z-axis, up and down rotates it around the X-axis and simultaneously depressing the right mouse button will rotate the view around the world's Y-axis.

The Viewing Angle is selected by default as this is the first change, if any, likely to be made to the display.

⦿ **Zoom** radio button ~ The Zoom control is the equivalent of the interactive ZOOM toolbar button (see page 1~18 in the I Files) in that it controls the scale at which the object is displayed in the Render window. The size at which the object is initially displayed is based upon a "bounding box" that encompasses all the points defining it.

By default, the Zoom option is not active, but when selected, placing the cursor over the Render window and dragging the mouse to the left will shrink the size of the object, whilst dragging it to the right will magnify the object.

⦿ **Perspective** radio button ~ The Perspective control is the equivalent of the interactive PERSPECTIVE toolbar button (see page 1~18 in the I Files) in that it controls the perspective convergence at which the object is displayed in the *Render window*.

By default, the Perspective option is not active, but when selected, placing the cursor over the *Render window* and dragging the mouse to the left compresses the perspective, whilst dragging to the right exaggerates the perspective of the object.

⦿ **Lighting Angle** radio button ~ The Lighting Angle option allows the position of the light illuminating the object(s) in the Render window to be varied. This is the equivalent control to the interactive LIGHTING toolbar button found in the *Perspective view* (see page 25).

By default, the Lighting Angle option is not active but when selected, wherever the left mouse button is depressed whilst the cursor is over the Render window determines the new position for the light source (see page 26).

Placing the cursor over the Render window and depressing the right mouse button displays a pop-up menu providing controls governing the display in the Render window.

Use OpenGL menu selection ~ Selecting this option uses the OpenGL rendering routines as described earlier (see page 22) using either hardware acceleration (where available) or software interpretation.

When the Use OpenGL is not selected the COLOR display mode currently active in the *Perspective view* is used, using either 236 or 24Bit/16M colours (see page 24).

Show Lines menu selection ~ The Show Lines option superimposes the object mesh over the OpenGL or 24 Bit display; the mesh being drawn in white. This option can be useful for visualising the geometric complexity across the object.

If any quick edges have been defined and the QUICKDRAW command is active, these will be used instead, otherwise the normal hidden line removal display of the mesh will be used.

Use B. Boxes menu selection ~ The Use B. Boxes option redraws the object according to the current quickdraw selection when the display in the Render window is being manipulated with the Viewing Angle, Zoom or Perspective options. Either a box encompassing the objects axes will be used or the currently defined quick edges; this will depend on the current status of the QUICKDRAW command in the *Editor*.

This option allows realtime manipulation of more complex objects in the Render Window.

The three previous options all default to the options currently active in the *Editor Perpsective view*. Of course, they can be varied independently from it.

❑ **OK** button ~ Selecting the OK button accepts the current changes to the object's complexity and closes the *Portal*, reinstating the *Editor* views with the optimised object.

❑ **Cancel** button ~ Selecting the Cancel button ignores any changes made and closes the *Portal*, reinstating the *Editor* views with the object's complexity remaining unaltered.

With the initial release of ***Imagine*** for Windows, selecting PRIMITIVE from the ***Object-New menu***, using the New PRIMITIVES toolbar button or selecting the ***F5*** keyboard shortcut, displayed the *Primitive Type dialog*. From this the appropriate *Properties dialog* was displayed for the chosen primitive allowing the various parameters to be set (see page 2~20 in the I Files). Having specified the size and the geometric complexity of the primitive it was placed at the centre of the three *Fixed views* highlighted, ready to be picked. Furthermore, because new primitives are only supplied with a default set of attributes, namely white and Phong Shaded (see page 5~1 in the I Files) the *Attributes dialog* then had to be displayed to specify more appropriate surfaces attributes. To streamline this process the following toolbar and "*Portal*" have been introduced…

Primitives drag and drop toolbar

This new toolbar streamlines the work flow inasmuch that all the pre-defined objects (including the special objects: axis, ground and perfect sphere) are now available by selecting the PRIMITIVE command and are placed in the *Editor* picked, ready for manipulation. Additionally, as will be discussed, the parameters controlling the size and shape of the primitive are included on a *Tab* within the *Attributes portal* (see page 94) thereby allowing an appropriate surface to be specified right from the start.

F5

Drag and Drop Primitives toolbar ~ ***Object-New Primitives*** ~ Selecting this command now displays the new *Primitives Drag and Drop toolbar*. This toolbar is slightly different to the others in that it cannot be "docked" to the sides of the *Editor*. The *Primitives Drag and Drop toolbar* is however a floating toolbar and as such can be moved to any convenient location on the screen and be resized in both the horizontal and vertical directions to suit the user's preference. In addition, selecting the right mouse button whilst the cursor is over the *Toolbar* displays a pop-up menu which provides the option to have a vertical or horizontal scrollbar. The toolbar can of course be closed.

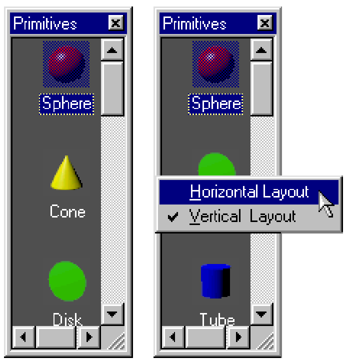
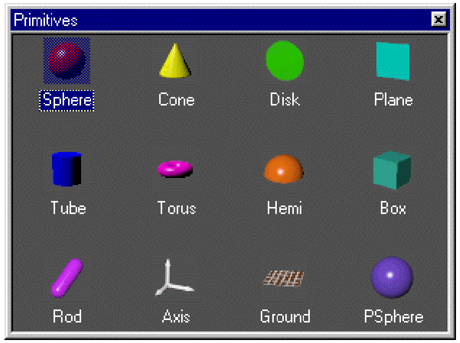


Figure 49: The default size and format for the Primitives Drag and Drop toolbar (left) with it's pop-up menu (right).

All the pre-defined objects (Sphere, Cone, Disk, Plane, Tube, Torus, Hemi, Box, Rod, Axis, Ground and perfect Sphere) are available directly from the toolbar.



There are two methods of selecting the required object: either by double-clicking the icon representing the primitive or by using the left mouse button to drag the icon from the *Toolbar* and drop it on one of the *Editor views*. Double-clicking the object will placed it in the centre of each of the three *Fixed views*, whereas dragging and dropping the object will place it in the *Editor* wherever the mouse button is released. The object will be positioned at the co-ordinates of the cursor (when the mouse button is released) in the active *View*. The position the object assumes in the third dimension not accommodated by the active view is, as might be expected, simply the point in the world co-ordinate system on which the *Views* are centred.

Figure 50: The Primitives Drag and Drop toolbar can be resized in both it's vertical and horizontal format to display some or all of the available primitives, and be positioned anywhere on the screen.

Ini file

The following entries in the "Imagine.ini" file allow the *Toolbar* to be automatically opened the next time ***Imagine*** is started, remembering both it's size and position.

`[PrimDlg]`

`Left=164`

> *`Toolbar`*`'s position from the side of the screen, in pixels`

`Top=248`

> *`Toolbar`*`'s position from the top of the screen, in pixels`

`Width=89`

The width of the *Toolbar*, in pixels

Height=243

The height of the *Toolbar*, in pixels

Visible=0

0= *Toolbar* is not displayed

1= *Toolbar* is displayed

Horizontal=0

0= Vertical scrollbar is displayed

1= Horizontal scrollbar is displayed

Primitives portal

The other major change from the old *Primitive Type dialog* is that all the controlling parameters for the specified primitive are now included on a *Shape tab*, which itself is part of the new *Attributes portal* (see page 94). The advantages to this method are twofold. Firstly, with the new Render window, this gives an instant visual feed-back on the parameters chosen and how they affect the object's size, complexity, geometry and edge treatment. The other distinct advantage is as a result of the *Shape tab* being part of the *Attributes portal*. Now, surface attributes can be specified as the object is created. This may directly influence the object's geometric complexity for, say, a fog object or bitmapped/textured object.

The specifics of how the new *Portal* options are controlled will be discussed later (see page 94). The only significant difference between this *Portal* and the *Attributes portal* itself (aside from the *Shape tab*) is the default settings. The *Primitive portal*, as might be expected, opens with the Object radio button selected; thus displaying the chosen primitive. This is not the case with the *Attributes portal*.

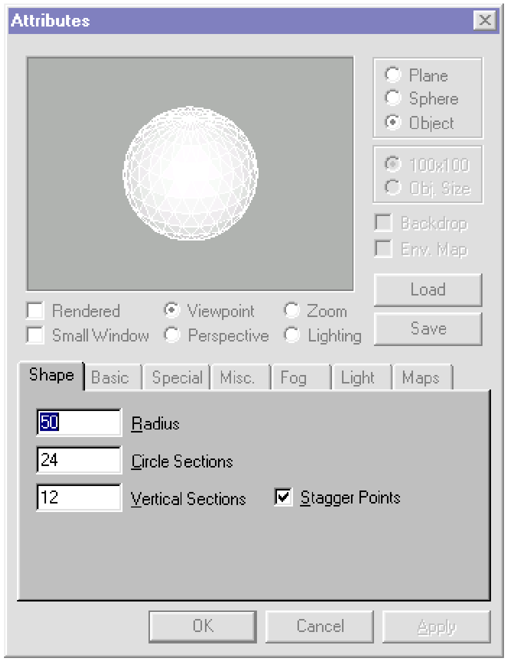


Figure 51: The Primitives portal is based on the Attributes portal, but includes a Shape tab.

Whilst virtually any object can be built based upon the existing primitives in conjunction with ***Imagine***'s extensive range of modelling tools, some objects require many tools to produce. Consequently, any assistance in the modelling process can only be welcomed; hence the inclusion of…

Additional Primitives

To coincide with the introduction of the *Primitives Drag and Drop toolbar*, three new object primitives have been added over the course of the Constant Upgrade Programme. However, before considering these, a couple of small but timesaving options have been added to the existing primitives.

Previously, when a "lid" and/or "base" was added to the Tube or Cone primitives, the edge defining this rim was rendered smooth. This was because all the edges comprising the default primitives were defined as being "soft edges". Generally, the edges defining the rim (joining the sides and the top/bottom) should be rendered as sharp edges for more conventional looking objects. This was a long-winded procedure involving picking the object, entering PICK EDGES mode (see page 2~50 in the I Files) selecting the edges and executing the MAKE SHARP EDGES command (see page 2~50 in the I Files).

Now, with the inclusion of the Sharp Edges option (selected by default) the Tube and Cone primitives are automatically created with "rims" that when rendered will appear sharp. This option does not affect the curved surfaces; they will continue to be treated as soft edges and rendered smoothly.

The other small enhancement concerns the Ground and Axis pre-defined special objects. Both of these can now have their axis size specified at the time of creation. This removes the need of resizing the axis from the default values using the TRANSFORMATION command (see page 2~65 in the I Files). This can be especially useful where, for example, an axis is being used as a light source; the falloff being controlled by the size of the Y-axis (see page 7~37 in the I Files).

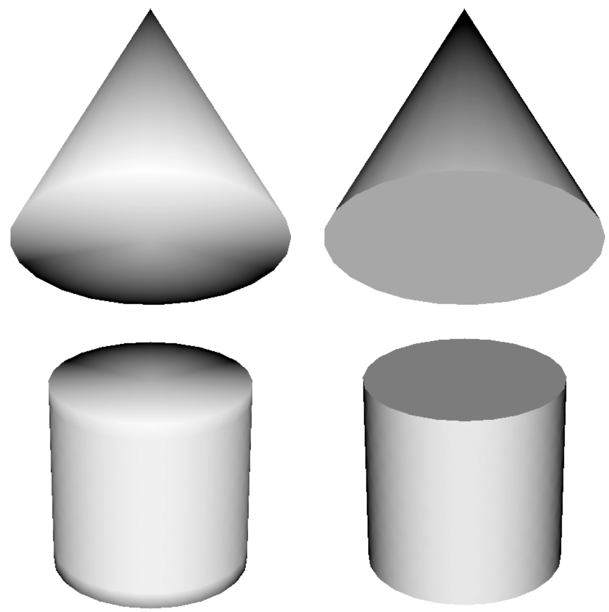


Figure 52: The results the Sharp Edges option has on the rim of a Cone (top) and a Cylinder (bottom) when rendered; selected (right) and deselected (left).

Returning to the additional primitives, it is worth pointing out that each of them could previously be modelled by manipulating the existing primitives. However, in accordance with the continuing theme of streamlining the use of all aspects in ***Imagine***, the following three primitives (composed of points, edges and faces) have been added. These, of course, can be used as objects in their own right or as the basis for ones that are more complex…

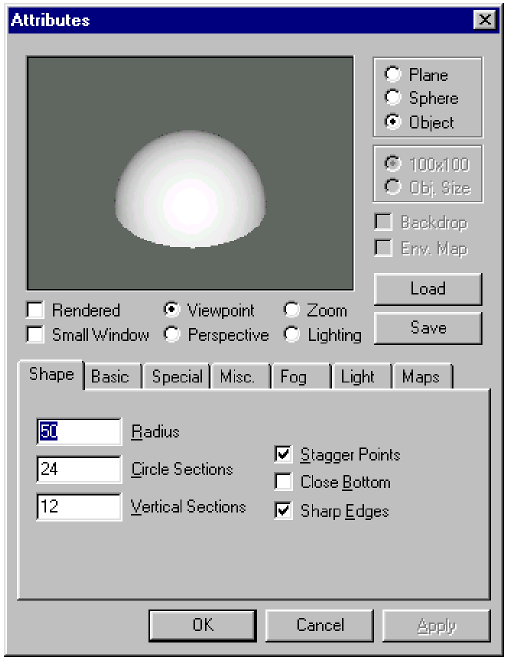


Figure 53: The Hemi primitive portal showing the default values on the Shape tab.

Hemi

The Hemi (short for hemisphere) Drag and Drop Primitive is, as might be expected, half a sphere, with options to determine whether it should be created as a "open-ended" or appear to be "solid".

Previously, modelling a hemisphere involved using the **SLICE** command (see page 2~74 in the I Files) on a Sphere primitive (see page 2~20 in the I Files) or moulding an arc. The following parameters — all located in the *Shape tab* controlling its size, complexity and edge treatment — rationalise the construction of this type of object considerably…

- **Radius** input box ~ The Radius parameter determines the radius of the hemisphere; the default value being 50 ***Imagine*** units.

Specifying a negative value will create a hemisphere of the desired size. However, when placed in the *Editor* the object will be "flipped" on all three axes.

- **Circle Sections** input box ~ The Circle Sections parameter determines the number of points used to define the circumference of each Circle Section.

The default value of 24 is a good compromise between speed and quality of rendering. The greater the number of points specified the smoother the circle and consequently the smoother the hemi-

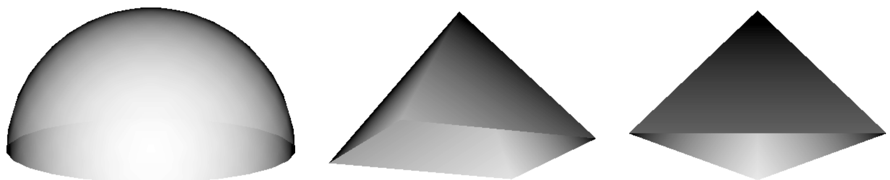


Figure 54: By specifying different values for the Circle Sections parameter, a variety of different objects can be created: Hemisphere (left) four sided pyramid (centre) and a three sided pyramid (right).

sphere will render, but at the expense of longer rendering times and increased use of system resources.

- ❑ **Vertical Sections** input box ~ The Vertical Sections parameter determines the number of Circle Sections used to create the hemisphere.

The default value of 12 is a good compromise between speed and quality of rendering. The greater the number of Vertical Sections specified the smoother the rendered profile of the hemisphere, but at the expense of longer rendering times and increased use of system resources.

- ☑ **Stagger Points** check box ~ Selecting the Stagger Points option causes the points forming each Circle Section to be offset to those in it's neighbouring Vertical Sections. Staggering them in this way allows the hemisphere to appear smooth when rendered whilst using fewer Circle and Vertical Sections.

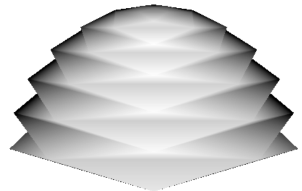


Figure 55: The Stagger Points option causes the hemisphere to render smoother. Deselecting it can in conjunction with only a few Circle Sections can produce some interesting looking objects.

By default, this option is selected and should generally be left as such. However, when the hemisphere is being used as the basis for another object and point editing is likely, it may be preferable to leave the points aligned.

- ☑ **Close Bottom** check box ~ Selecting the Close Bottom option will produce a "solid" rather than the default "hollow" hemisphere by automatically adding faces around the rim.

Generally, when this option is selected the following option should also be used to prevent the rim separating the curved sides and flat bottom being rendered smoothly.

- ☑ **Sharp Edges** check box ~ Selected by default, the Sharp Edges option will ensure the edges forming the rim separating the sides from the bottom will be rendered as sharp. This of course, assumes that the previous option has been selected to produce a "solid" hemisphere.

This option will only affect those edges forming the rim; the curved surfaces will still be rendered smoothly. Deselecting this option will cause ***Imagine*** to try to render the rim smoothly. At close quarters this may look slightly

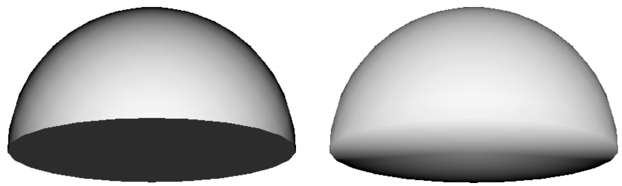


Figure 56: The Sharp Edges option defines the rim to be rendered as sharp (left). Deselected, Imagine attempts to render this rim as though there is a smooth transition between the flat base and the curve of the hemisphere.

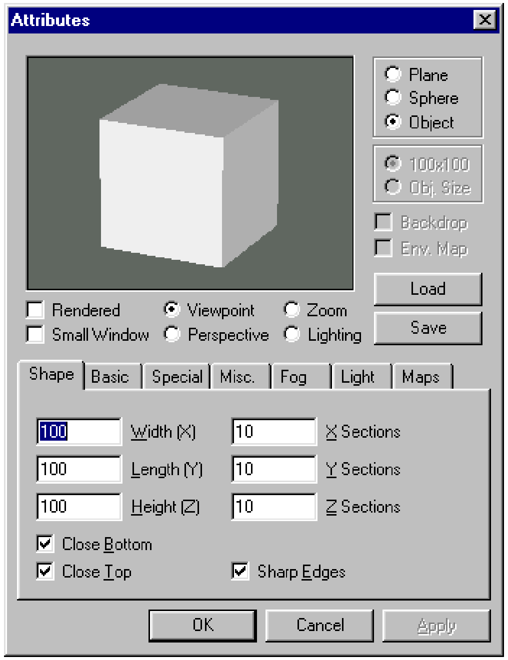


Figure 57: The Box primitive portal displaying the Shape tab and its default settings.

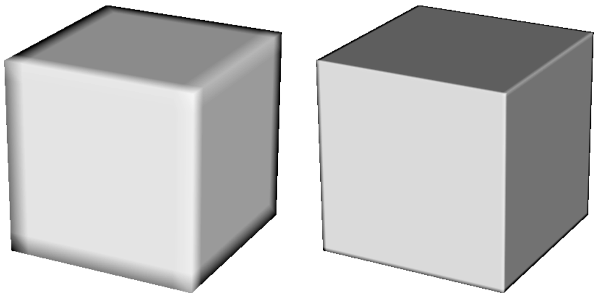


Figure 58: The default box with Sharp Edges deselected (left). Increasing the number of sections to 50 produces smoother, soft edges (right).

odd in that there will be a smooth transition in shading between the curved sides and flat bottom whilst the rim will still appear sharp in profile, as will it's cast shadow.

Box

The Box Drag and Drop Primitive not surprisingly creates a cube, which can be "open-ended" or appear to be "solid".

Previously, modelling a cube involved creating a plane, picking it and then extruding it. The following parameters — all located on the *Shape tab* controlling its size, complexity and edge treatment — rationalise the construction of this type of object considerably…

- **Width (X)** input box ~ The Width parameter specifies the width (or X dimension) of the box, the default value being 100 ***Imagine*** units.
- **Length (Y)** input box ~ The Length parameter specifies the length (or Y dimension) of the box, the default value being 100 ***Imagine*** units.
- **Height (Z)** input box ~ The Height parameter specifies the height (or Z dimension) of the box, the default value being 100 ***Imagine*** units.
- **X Sections** input box ~ The X Sections parameter specifies the number of equal sections by which the box will be subdivided across it's width (or X dimension).

  The default value generates a box composed of 10 sections across its width. Whilst this may initially seem overly complex for a plain box (perhaps the minimum value of 3 would be more appropriate) where the Box primitive will be used as the basis for a different object, the additional sections will allow for smoother deformations.

- **Y Sections** input box ~ The Y Sections parameter specifies the number of equal sections by which the box will be subdivided along its length (or Y dimension).

The default value of 10 sections allows for smoother deformations, whilst the minimum value of 3 might be more appropriate for a simple cube.

❑ **Z Sections** input box ~ The Z Sections parameter specifies the number of equal sections by which the box will be subdivided along its height (or Z dimension).

The default value of 10 sections allows for smoother deformations, whilst the minimum value of 3 might be more appropriate for a simple cube.

☑ **Close Button** check box ~ Selecting this option will "close" the base of the box by automatically adding faces to the bottom of the box.

☑ **Close Top** check box ~ Selecting this option will "close" the lid of the box by automatically adding faces to the top of the box.

Combining both of these options will produce a "solid" box rather than a "hollow" or open-ended one.

☑ **Sharp Edges** check box ~ Selecting the Sharp Edges option will ensure that the edges forming the four verticals and those defining the rims separating the sides from the top and/or bottom will be rendered as sharp. This, of course, assumes that either one or both of the options in the previous two parameters have been selected to produce a "solid" or "open-ended" box.

Sharp Edges is selected by default thus rendering the box with sharp edges. When deselected, ***Imagine*** will attempt to render these edges as soft. However, this does not alter the rectangular outline of the box, consequently the effect is not so successful when viewed at close quarters. Using the SMOOTHING TOOL command with the Lock Perimeter Points option will greatly improve this situation by actually smoothing the geometry to match (see page 2~87 in the I Files).

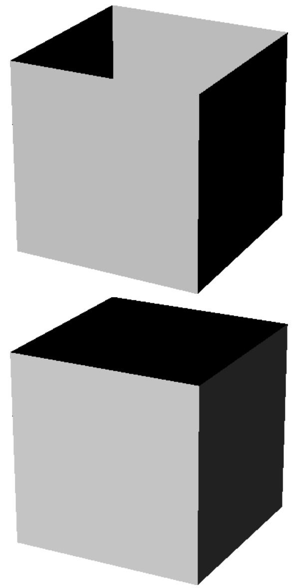


Figure 59: With the Sharp Edges option selected, the box can appear open-ended (top) or closed (bottom) using the Close Bottom and Top options.

Rod

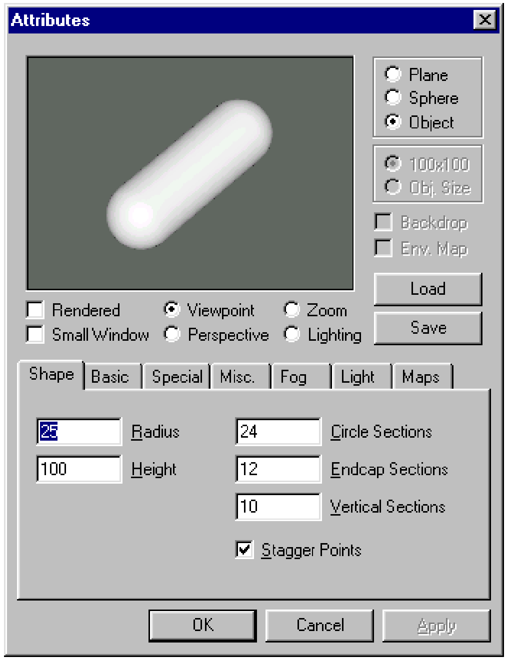


Figure 60: The Rod primitive portal displaying the Shape tab and its default settings.

The Rod Drag and Drop Primitive is the combination of a Tube primitive with a Hemi primitive at each end.

Previously, one way of producing this type of object required a Sphere primitive with half of the points deleted, the resulting hemisphere copied, pasted, translated the required amount, joined and then the **SET EDGE LINE** and **FILL TO EDGE LINE** commands (see page 2~53) used to add a skin between the two. The following parameters — all located on the *Shape tab* controlling its size and complexity — streamline the construction of this type of object considerably…

❑ **Radius** input box ~ The Radius parameter determines the radius of the tube and hemisphere components that combine to make the Rod, the default value being 25 ***Imagine*** units.

Specifying a negative value will create a Rod of the desired size, however, when placed in the *Editor* the object and its axes will be “flipped” in relation to the world co-ordinate system.

❑ **Height** input box ~ The Height parameter specifies the height (or the Z dimension) of the Rod, the default value being 100 ***Imagine*** units.

The height in this instance refers to the tube portion of the Rod; the value specified for the Radius will be added to the height value.

❑ **Circle Sections** input box ~ The Circle Sections parameter determines the number of points used to define the circumference of each Circle Section.

The default value of 24 is a good compromise between speed and quality of rendering. The greater the number of points specified the smoother the circle and consequently the smoother the Rod will be rendered.

❑ **Endcap Sections** input box ~ The Endcap Sections parameter determines the number of Circle Sections used to define each of the hemispherical ends.

The default value of 12 is a good compromise between speed and quality of rendering. The greater the number specified the smoother the rendered profile of each end, but at the expense of longer rendering times and increased use of system resources.

- ❑ **Vertical Sections** input box ~ The Vertical Sections parameter determines the number of Circle Sections used to create the cylindrical part of the Rod.

The default value of 10 allows smooth deformations to be subsequently performed. Where the Rod is destined to remain “rod-shaped”, this value could be reduced to 1 to conserve system resources and minimise rendering time.

- ☑ **Stagger Points** check box ~ Selecting the Stagger Points option causes the points forming each Circle Section to be offset to those in it’s neighbouring Vertical and Endcap Sections. Staggering them in this way allows the Rod to appear smooth when rendered, whilst specifying fewer Circle, Endcap and Vertical Sections.

By default, this option is selected and should generally be left as such. However, when the Rod is being used as the basis of another object where point editing is likely, it may be preferable to leave the points aligned.

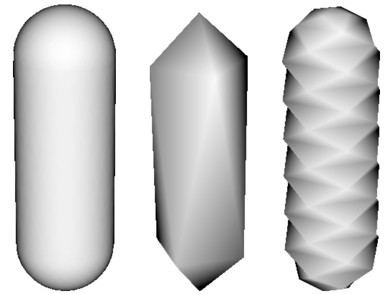


Figure 61: The default Rod (left) the effect of using smaller values for the Circle Sections (middle) and combined with deselecting the Stagger Points option (right).

As was indicated earlier, the *Portal* displayed following the selection of one of the objects from the new Primitives Drag and Drop toolbar, is based on the…

Attributes portal

The only significant difference between the two is the lack of a *Shape tab*, which as may be recalled, contains all the parameters for controlling the geometry of the object.

F7

Attributes ~ ***Functions menu*** ~ Selecting the **ATTRIBUTES** command displays the new *Attributes portal* controlling the surface appearance of the picked object.

Fundamentally, the *Attributes portal* is very similar to its predecessor, the *Attributes dialog*. All the parameters on the six *Tabs* are identical (see Chapter 5 in the I Files). The only significant change is to the viewing options controlling the *Portal* itself.

Following the implementation of the mini "perspective view" in the Render window and its associated interactive viewing controls, it was felt that the view alignment controls on the old *Dialog* were no longer necessary. Consequently, the Front, Back, Right, Left, Top and Bottom option buttons and the Align Object to World check box have all been removed.

However, this does mean that when checking the alignment of brushmaps and textures that are not orientated to the world co-ordinate system, the view of the object needs to be interactively changed in the Render window (see below). Furthermore, since the Plane and Sphere options have been optimised for speed of display, the interactive viewing control is unavailable. Consequently, this requires that any alignment of brushmaps and textures to anything other than the X-Z plane be performed using the Object option.

The Plane, Sphere and Object (see page 5~4 in the I Files) the Backdrop and Env. Map check boxes (see pages 5~5 and 5~6 in the I Files) the Load, Save, OK, Cancel and Apply buttons (see page 5~3 in the I Files) all perform as they did in the old *Attributes dialog*.

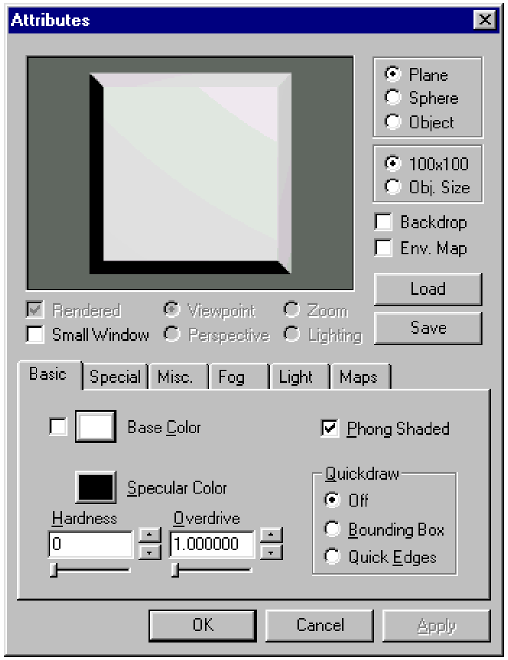


Figure 62: The Attributes portal displaying the Basic tab with its default settings.

The new options are geared towards controlling the display in the Render window…

☑ **Rendered** check box ~ The Rendered option determines that the display is actually rendered using either scanline or ray-trace. Which of the two rendering methods is used will depend on the current selection in the *Render mode menu* (see below). Although access to this option is not available whilst the Plane or Sphere object types are active, the display in the Render window will be rendered.

The Render option is selected by default, unlike the *Portal* displayed when adding a new primitive. This is because rendering takes all of the more advanced attributes settings such as reflectivity, transparency, fog and the application of brushmaps and/or textures into account . However, with more complex objects or objects with many brushmaps/textures applied to them, the responsiveness of display may lag behind. By deselecting the Rendered option (whilst the Object option is active; this can then be changed back to Plane or Sphere) the calculations involved in displaying the object are greatly simplified and therefore the Render window becomes more responsive. This is because one of the more “basic” display methods will be used, 24BIT/16M COLORS or OPENGL again, depending on the current selection in the *Render mode menu*. As you may recall, none of these display methods takes into account the more “complex” surface attributes, calculating only Base Color, Specular Color and Hardness.

☑ **Small Window** check box ~ The Small Window option allows the display to be even more responsive by reducing the size of the Render window by half. This option is especially useful for less powerful computers and where complex or objects with many layers of brushmaps and/or textures are being manipulated.

Initially, this is the only option available when the *Attributes portal* is first displayed and remains available at all times. By default, the Small Window option is not selected, allowing the *Render window* to display at its maximum size.

⦿ **Viewpoint** option button ~ The Viewpoint option is the equivalent of the interactive ANGLE toolbar button (see page 1~18 in the I Files) in that it controls the orientation of the object in the Render window. Depressing the left mouse button whilst the cursor is over the Render window will cause the object to be redrawn according to the currently active display option (see below) thus allowing real-time feedback as the view is manipulated. Dragging the cursor from left to right rotates the view around the World's Z-axis, up and down rotates it around the X-axis and simultaneously depressing the right mouse button will rotate the view around the world's Y-axis.

The Viewing Angle is selected by default as this is the first change, if any, likely to be made to the display.

⦿ **Perspective** radio button ~ The Perspective control is the equivalent of the interactive PERSPECTIVE toolbar button (see page 1~18 in the I Files) in that it controls the perspective convergence at which the object is displayed in the Render window.

By default, the Perspective option is not active, but when selected, placing the cursor over the Render window and dragging the mouse to the left compresses the perspective, whilst dragging to the right exaggerates the perspective of the object.

⦿ **Zoom** radio button ~ The Zoom control is the equivalent of the interactive ZOOM toolbar button (see page 1~18 in the I Files) in that it controls the scale at which the object is displayed in the Render window. The size at which the object is initially displayed is based upon a "bounding box" that encompasses all the points defining that.

By default, the Zoom is not active, but when selected, placing the cursor over the Render window and dragging the mouse to the left will shrink the size of the object, whilst dragging it to the right will magnify the object.

⦿ **Lighting Angle** radio button ~ The Lighting Angle option allows the position of the light illuminating the object(s) in the Render window to be varied; whether the Rendered option is selected or not. This is the equivalent control to the interactive LIGHTING toolbar button found in the *Perspective view* (see page 25).

By default, the Lighting Angle option is not active but when selected, wherever the left mouse button is depressed whilst the cursor is over the *Render window* determines the new position for the light source (see page 26).

By depressing the right mouse button whilst the cursor is over the Render window, a menu is revealed providing options controlling the render type and how the objects are to be displayed; including when the view is being manipulated…

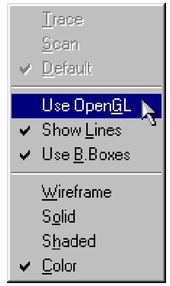


Figure 63: The pop-up menu displayed when the right mouse button is selected over the Attributes portal Render window.

The first group of three: Trace, Scan and Default, are the familiar options carried over from the older *Attributes dialog* (see page 5~5 in the I Files) and determine how the display in the *Render window* is rendered. Briefly, scanline rendering approximates transparent surfaces, certain types of reflection and any brushmaps and/or textures. Raytracing provides the best quality of rendering, including accurately calculating transparent and reflective surfaces but at the expense of display time. The Default option, which is initially selected, uses whatever render option is currently active in the *Quickrender Settings dialog* (providing this is either scanline or raytrace); initially a quickrender defaults to scanline rendering.

These options are only available for modification when the Rendered check box is selected (see page 95). Conversely, when the Rendered option is deselected, the second group of seven options becomes available…

Use OpenGL menu selection ~ Selecting this option uses the OpenGL rendering routines as described earlier (see page 22) using either hardware acceleration (where available) or software interpretation.

By default, this option is deselected causing the display to be rendered using the 24 bit/16M Colors render type.

Show Lines menu selection ~ The Show Lines option — active by default — superimposes the object mesh over the OpenGL or 24 bit/16M display; the mesh being drawn in white. This option can be useful for visualising the geometric complexity across the object.

If any quick edges have been defined and the QUICKDRAW command is active, these will be used instead, otherwise the normal hidden line removal display of the mesh will be used.

Use B. Boxes menu selection ~ The Use B. Boxes option — active by default — redraws the object according to the current quickdraw selection when the display in the Render window is being manipulated with the Viewpoint, Perspective or Zoom options. Either a box encompassing the object's axes will be used or the currently defined quick edges; this will depend on the current status of the QUICKDRAW command in the *Editor*.

This option allows realtime manipulation of more complex objects in the Render Window.

Wireframe menu selection ~ The Wireframe display mode shows all of the edges/faces defining the object.

This option is useful for its responsiveness and speed of display as well as providing some idea of the objects geometric complexity.

Solid menu selection ~ The Solid display mode takes marginally longer to display due to the hidden line removal calculations performed.

Hidden line removal simplifies the display by not drawing those edges/faces obscured by the "front" of the object.

Shaded menu selection ~ The Shaded display mode refines the display in the Render window further by using 216 levels of grey flat shaded or 8 bit Gouraud shading when the OpenGL option is active.

With this option selected the form of the object is apparent, but at the expense of slightly longer redraw times.

Color menu selection ~ The Color display option is the best level of display offered, using 216 colours flat shaded or 24 bit Gouraud shading when the OpenGL option is active.

Continuing with the theme of ensuring the more complex options within ***Imagine*** and the effect they are going to have are comprehensible, the *F/X dialogs* have also received the "*portal*" treatment…

F/X Portals

Implementing each F/X as a *Portal* provides a visual representation of how varying each F/X's parameter will affect the results on the object to which it is applied. Compare this to the previous state of affairs where a parameter was varied "blind", the result only being calculated after the *Dialog* was closed and the *Editor view* reinstated. Consequently, this allows a more intuitive method of fine-tuning the available parameters to achieve precisely the desired result.

It should be stressed that some of the Stage F/X, especially the global ones are not suited to the way objects are previewed in the *Portal* Render window.

Each Stage F/X — both Object and Global — still maintains it's parameters arranged in a logical fashion across one or more *Tabs*, but these *Tabs* are now arranged below a *Render window* with the following display controls…

- **Viewing Angle** radio button ~ The Viewing Angle option is the equivalent of the interactive ANGLE toolbar button (see page 1~18 in the I Files) in that it controls the orientation of the object in the Render window. Depressing the left mouse button whilst the cursor is over the Render window will cause the object to be redrawn according to the currently active display option (see below) thus allowing real-time feedback as the view is manipulated. Dragging the cursor from left to right rotates the view around the World's Z-axis, up and down rotates it around the X-axis and simultaneously depressing the right mouse button will rotate the view around the world's Y-axis.

The Viewing Angle is selected by default as this is the first change, if any, likely to be made to the display.

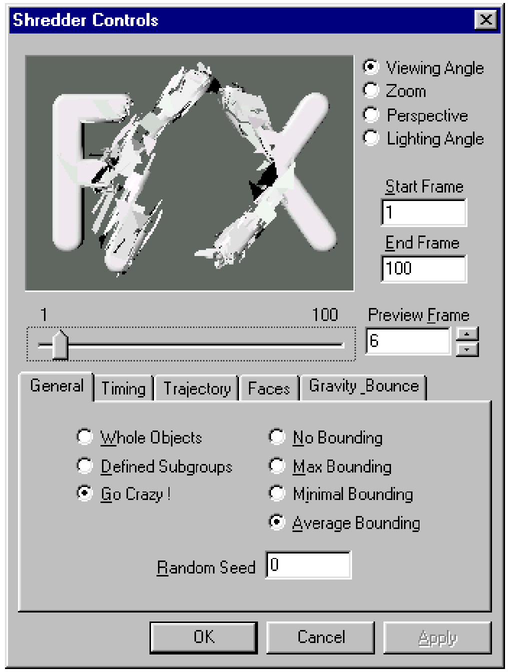


Figure 64: The "real-time" display in the F/X portals; in this case the Shredder F/X.

⦿ **Zoom** radio button ~ The Zoom control is the equivalent of the interactive ZOOM toolbar button (see page 1~18 in the I Files) in that it controls the scale at which the object is displayed in the Render window. The size at which the object is initially displayed is based upon a “bounding box” that encompasses all the points defining that object (or all the axes defining a path for a path object, or the size of the axis itself for an object containing no points).

By default, the Zoom option is not active, but when selected, placing the cursor over the Render window and dragging the mouse to the left will shrink the size of the object, whilst dragging it to the right will magnify the object.

⦿ **Perspective** radio button ~ The Perspective control is the equivalent of the interactive PERSPECTIVE toolbar button (see page 1~18 in the I Files) in that it controls the perspective convergence at which the object is displayed in the Render window.

By default, the Perspective option is not active, but when selected, placing the cursor over the Render window and dragging the mouse to the left compresses the perspective, whilst dragging to the right exaggerates the perspective of the object. Initially, the amount of perspective converge is derived from the amount currently set for the *Perspective view*. This can, of course, be varied independently from the *Perspective view* by using this option.

⦿ **Lighting Angle** radio button ~ The Lighting Angle option allows the position of the light illuminating the object(s) in the *Render window* to be varied. This is the equivalent control to the interactive LIGHTING toolbar button found in the *Perspective view* (see page 25). Wherever the left mouse button is depressed whilst the cursor is over the Render window determines the new position for the light source (see page 26).

☐ **Start Frame** input box ~ The Start Frame parameter determines at which frame in the animation the chosen Stage F/X is to act upon the object to which it is applied.

The initial frame displayed will depend on both where and how the F/X is applied. If a Stage F/X is applied from within the *Detail editor* using the APPLY STAGE F/X command (see page 2~87) then the Start Frame will default to 1; the first frame in the “temporary” animation. If the Stage F/X is applied from the *Action dialog* in the *Stage editor*, the first frame selected when the F/X bar is added (either double-clicked or drag selected, see page 7~ 120) will be the one displayed.

☐ **End Frame** input box ~ The End Frame parameter determines at which frame in an animation the chosen Stage F/X finishes its influence over the object to which it is applied.

The initial frame displayed will depend on both where and how the F/X is applied. If a Stage F/X is applied from within the *Detail editor* using the APPLY STAGE F/X command (see page 2~87) then the End Frame will default to 100; the last frame in the “temporary” animation. If the Stage F/X is applied from the *Action dialog* in the *Stage editor*, the last frame selected when the F/X bar is added (either double-clicked or drag selected, see page 7~ 120) will be the one displayed. It is highly recommended that the drag selection method is used, otherwise there will be no frames initially registered for the F/X to be previewed; the *Portal* will have to be accepted and re-opened.

⇔ **Preview Frame** slider ~ The Preview frame slider, input box and “spinners” allows the influence the chosen Stage F/X has over the object to be previewed at a particular frame within the maximum number of frames specified.

Depressing the right mouse button whilst the cursor is over the Render window displays a pop-up menu providing the following options governing the display in the Render window…

Use OpenGL menu selection ~ Selecting this option uses the OpenGL rendering routines as described earlier (see page 22) using either hardware acceleration (where available) or software interpretation.

By default, this option is deselected causing the display to be rendered using the 24BIT/16M COLORS display type (see above).

Show Lines menu selection ~ The Show Lines option — active by default — superimposes the object mesh over the OpenGL or 24Bit/16M display; the mesh being drawn in white. This option can be useful for visualising the geometric complexity across the object.

If any quick edges have been defined and the QUICKDRAW command is active, these will be used instead, otherwise the normal hidden line removal display of the mesh will be used.

Use B. Boxes menu selection ~ The Use B. Boxes option — active by default — redraws the object according to the current quickdraw selection whilst the display in the Render window is being manipulated with the Viewing Angle, Zoom or Perspective display modifiers. Either a box encompassing the objects axes will be used or the currently defined quick edges; this will depend on the current status of the QUICKDRAW command in the *Editor*.

The following group of four options available from the menu all determine the quality of display shown in the Render window…

Wireframe menu selection ~ The Wireframe display mode shows all of the edges/faces defining the object.

This option is useful for its responsiveness and speed of display as well as providing some idea of the objects geometric complexity.

Solid menu selection ~ The Solid display mode takes marginally longer to display due to the hidden line removal calculations performed.

Hidden line removal simplifies the display by not drawing those edges/faces obscured by the “front” of the object.

Shaded menu selection ~ The Shaded display mode refines the display in the Render window further by using 216 levels of grey flat shaded or 8 bit Gouraud shading when the OpenGL option is active.

With this option selected the form of the object is apparent, but at the expense of slightly longer redraw times.

- **Color** menu selection ~ The Color display option is the best level of display offered, using 216 colours flat shaded or 24 bit Gouraud shading when the **OpenGL** option is active.

- ❑ **OK** button ~ Selecting the OK button accepts the parameters set in the *Portal*, closes it and reinstates the *Editor* display. When the Stage F/X is applied from within the *Detail editor* the object will be manipulated according the percentage of the effect set with the Preview Frame parameter.

- ❑ **Cancel** button ~ Selecting the Cancel button ignores any parameters set in the *Portal* and reinstates the *Editor* display with the object/group unchanged.

- ❑ **Apply** button ~ Selecting the Apply button implements any changes made to the options and parameters in the *Portal* and updates the object/group in the Render window to reflect these changes; the *Portal* remains open.

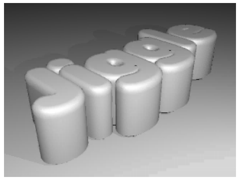


Figure 65: The Jiggle F/X with the default settings.

The Jiggle F/X section is dedicated to Michael, Becca and Julia Sherman who, I have heard, enjoy watching all the embedded AVI's in the I Files; move the cursor over the pics kids!

Whilst on the subject of Stage F/X, a new one was developed during the course of the Constant Upgrade Programme…

Jiggle F/X

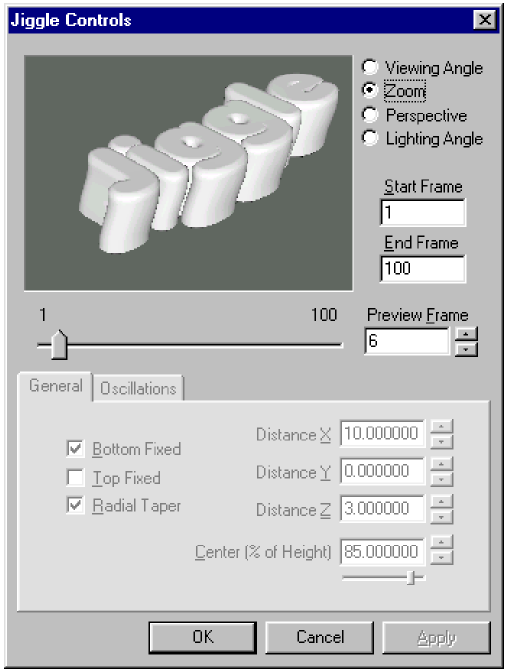


Figure 66: The Jiggle F/X portal showing the "real-time" display and how the chosen parameters affect the object to which it is applied.

The ***Jiggle.ifx*** causes the object to which it is applied to adopt an oscillating motion. The parameters available allow the object's material composition to be suggested and a variety of convincing motion types produced. Such things as a plate of wobbling jelly, curtains swaying in the breeze and an undulating waterbed can be easily recreated with the Jiggle Stage F/X.

The results of this F/X are in essence very similar to the smooth **SHEAR** deformation tool (see page 2~126 in the I Files) except in two important areas. Firstly, the Jiggle Stage F/X can be applied to both objects and groups of objects and secondly, it does not rely on the object/group axes. Unusually for tools within ***Imagine***, the size, position and orientation of the object/group axes are not considered; Jiggle uses the absolute world co-ordinate system. In effect, it creates a bounding box which encompasses all the points comprising the object/group, aligned to the world axes. Consequently, if an object is tilted in the *Stage editor* so that it's side is resting on the ground, as far as this F/X is concerned, the side becomes the bottom.

Selecting the ***Jiggle.ifx*** from either the *Action Dialog* (*Stage editor*) or the *Detail editor* (using the **APPLY STAGE F/X**, see page 2~87 in the I Files) displays the *Jiggle Controls portal* with the following parameters logically sorted across two *Tabs*…

General tab

☑ **Bottom Fixed** check box ~ Selected by default, the Bottom Fixed option causes the displacement to travel "up through" the object/group, leaving it's lowest point anchored in position.

Remember that this is the lowest point in relation to the world's Z-axis, which may not necessarily coincide with the object's base.

☑ **Top Fixed** check box ~ Selecting the Top Fixed option causes the displacement to travel "down through" the object/group, leaving it's uppermost point anchored in position.

Remember that this is the highest point in relation to the world's Z-axis, which may not necessarily coincide with the object's top.

Selecting both the Bottom and Top Fixed options cause the object/group to be anchored at both the lowest and highest points. Consequently, only the intervening portion can be displaced, producing an effect much like a belly dancer. Selecting neither of the two previous options causes the entire object/group to be displaced without any point being anchored.

☑ **Radial Taper** check box ~ Selected by default, the Radial Taper option causes the amplitude of the displacement to diminish the further from the object axis (or the parent object axis for groups) it travels; diminishing to no effect at the extremities of the object/group.

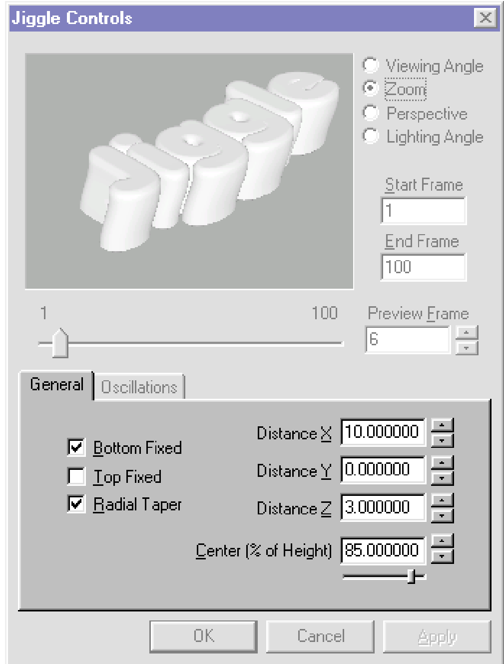


Figure 67: The Jiggle F/X portal showing the General tab with its default settings.

The Radial Taper only diminishes the jiggle around the world Z-axis and is confined to motion along this axis, i.e. up and down. Diminishing the Jiggle along the X and Y axes is not possible as in these cases it would appear "strange" for parts of an object close to the axis to be displaced whilst the extremities remain fixed.

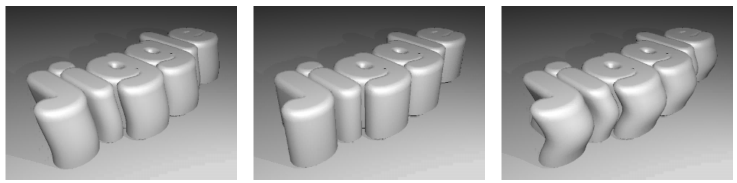


Figure 68: The results of varying the anchor point, with the Top Fixed option (left) the default Bottom Fixed option (centre) and both Top and Bottom Fixed options selected (right).

☐ **Distance X** input box ~ The value specified for the Distance X parameter determines the maximum distance in

Imagine units that a point determined by the Center parameter will be displaced along the world's X-axis; i.e. side to side.

The default value of 10 ***Imagine*** units provides an acceptable displacement for a "standard" sized object. Specifying negative values will cause the displacement to start along the world's negative X-axis before returning to the positive half.

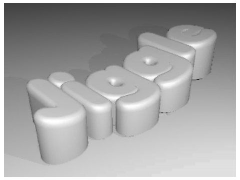


Figure 69: The results of the Radial Taper option.

☐ **Distance Y** input box ~ The value specified for the Distance Y parameter determines the maximum distance in ***Imagine*** units that a point determined by the Center parameter will be displaced along the world's Y-axis; i.e. back to front.

The default value of 0 ***Imagine*** units produces no displacement along the world's Y-axis. Specifying negative values will cause the displacement to start along the world's negative Y-axis before returning to the positive half.

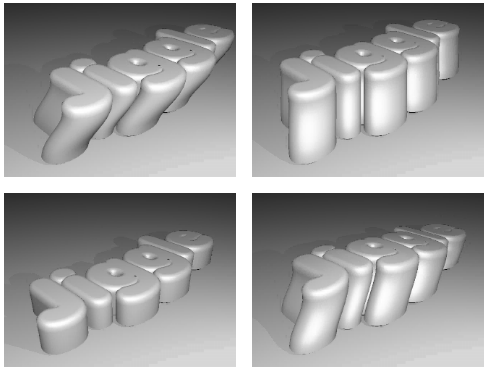


Figure 70: The effect of a 25 unit displacement along each of the axes; Distance X (top left) Distance Y (top right) Distance Z (bottom left) and all three (bottom right).

☐ **Distance Z** input box ~ The value specified for the Distance Z parameter determines the maximum distance in ***Imagine*** units that a point determined by the Center parameter will be displaced along the world's Z-axis; i.e. top to bottom.

The default value of 3 ***Imagine*** units provides just enough displacement in conjunction with the Distance X parameter to enhance a wobbling motion. Specifying negative values will cause the displacement to start along the world's negative Z-axis before returning to the positive half.

☐ **Center (% of Height)** input box ~ The Center parameter determines the position on the object around which the displacement is centred and at it's maximum.

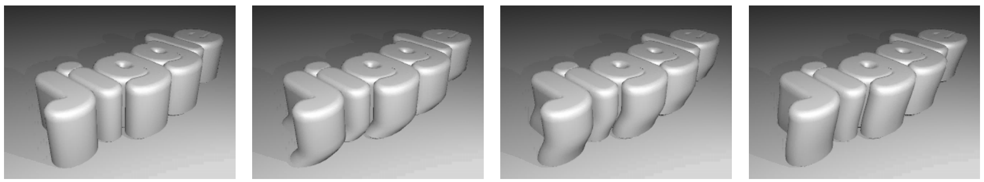


Figure 71: The results of varying the Center (% of Height) parameter; 0 (left) 25 (centre left) 50 (centre right) and 100 (right).

With the default value of 85%, the displacement is at it's maximum towards the top of the object (Bottom Fixed) with a smooth "shear" up to this point. The remaining 15% of the object/group will continue from where the "shearing" effect finishes with no further distortion. A value of 100% will cause the geometry at the extremity of the axis chosen to be displaced to the maximum, producing a smooth "shearing" of the object/group (assuming sufficient intervening geometry) along the entire length. A value of 0% will move the entire object/group along the chosen axis, but with no "shearing" effect.

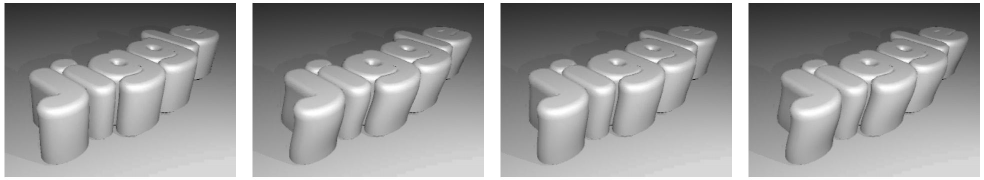


Figure 72: The results of varying the Oscillation Count parameter; 2.5 (left) 10 (centre left) 20 (centre right) and 40 (right).

Where the Top Fixed parameter is active the reverse is true, so 100% will cause the entire object/group to displace whilst 0% will cause the entire object/group to "shear".

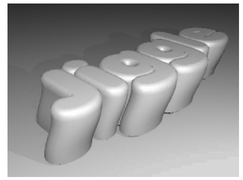


Figure 73: The result of of deselecting the Decaying Oscillations option.

Oscillations tab

☐ **Oscillation Count** input box ~ The Oscillation Count parameter determines the number of complete cycles (that is, displaced one way then the other and back) that will be applied over the duration of the *Jiggle.ifx*.

The default value of 5 creates a gentle wobble over the course of the standard 100 frame animation. Specifying smaller values will result in slower swaying motion, whilst higher values will create faster quivering type effects.

☑ **Decaying Oscillations** check box ~ Selected by default, the Decaying Oscillations option causes the amplitude of the displacement to gradually diminish to zero over the duration of the *Jiggle.ifx*.

Deselecting this option causes the amplitude of the displacement to remain constant for the duration of the *Jiggle.ifx*.

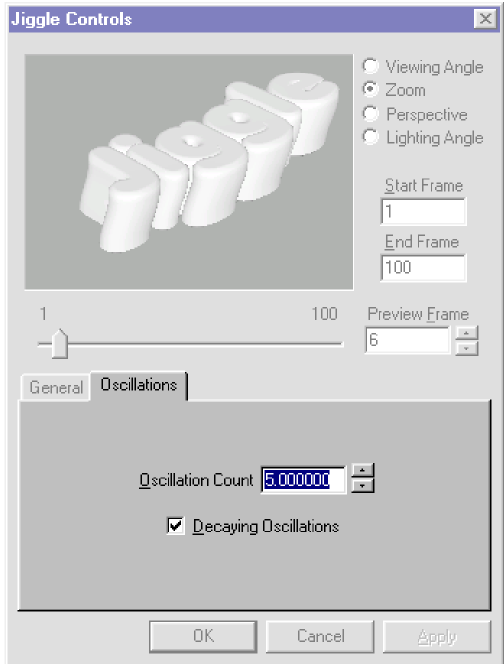


Figure 74: The Jiggle F/X portal showing the Oscillations tab with its default settings.

The final improvement made over the course of the "Constant Upgrade Programme" that remains to be considered is a simple improvement to an existing command that enhances it considerably…

enhanced Pick Subgroup

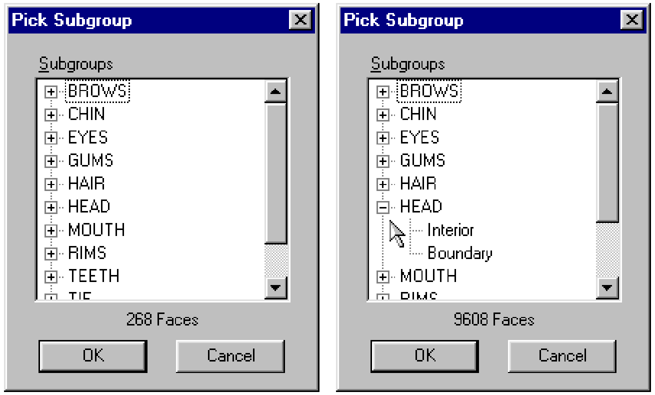


Figure 75: The Pick Subgroup dialog reporting the number of faces for the first subgroup in the list when none are selected (left) and revealing the Interior and Boundary options when the (+) symbol is selected (right).

As you make recall, the **Pick SUBGROUP** command (see page 2~60 in the I Files) becomes available whenever the **Pick POINTS**, **Pick EDGES** or **Pick FACES** editing modes are active, and where at least one subgroup of faces has been defined.

The "tree-based" *Pick Subgroup dialog* is virtually the same as the *Subgroup List dialog* it replaces except in a couple of important respects.

Firstly, there is now the option to not only pick the entire subgroup of points/edges/faces, but also the Boundary or Interior points/edges that form the subgroup — only the entire subgroup of faces can be picked. To reveal these two options the (+) symbol next to the subgroup name should be selected. Selecting either the Interior or Boundary options will pick only those points/edges that conform to the following rules:

- The Boundary edges are considered to be those edges that are used by exactly one face in the subgroup.
- The Interior edges are considered to be the ones used by two or more faces in the subgroup.
- The Boundary points are considered to be those points used by the Boundary edges.
- The Interior points are considered to be the points that are not Boundary points, but are used by some face in the subgroup.

Selecting the (-) symbol that replaces the (+) symbol will "close" the hierarchy.

Selecting the subgroup name will pick both the Interior and Boundary points/edges or all the faces. Multi-picking subgroups, Interior or Boundary items (or combinations thereof) is allowed by using the keyboard shortcuts ***Shift***, for picking consecutive items or ***Ctrl***, for picking items that do not appear next to one another. Even the Boundary items from one subgroup can be picked along with the Interior items from another. If points/edges/faces are already picked in the *Editor*, picking additional items from the *Pick Subgroup dialog* will add to those already picked.

As each subgroup, Interior or Boundary items are picked the number of faces associated with the picked item is displayed towards the bottom of the *Dialog*. Initially, the number of faces reported, even though no subgroups have been selected, refers to the first subgroup in the list.

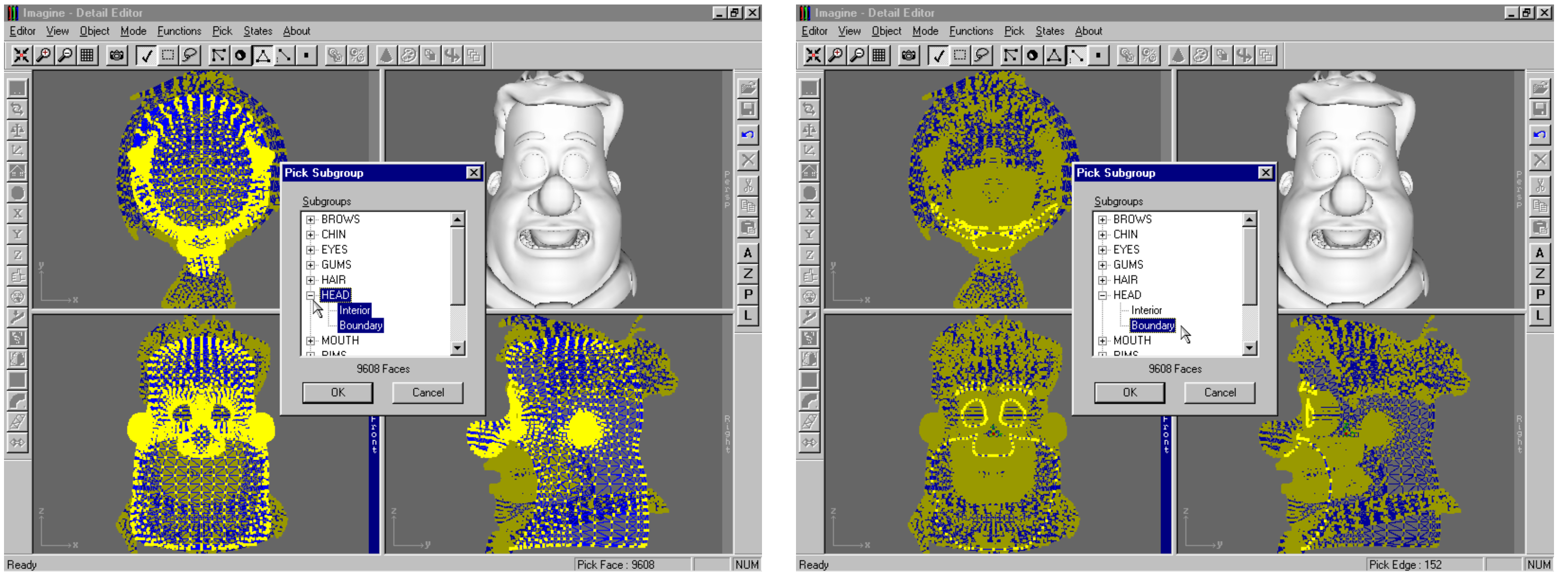


Figure 76: When using the Pick Subgroup command, only the entire subgroup of faces can be picked (left). When used with the Pick Edges mode either the Interior or Boundary edges forming the subgroup can be individually selected (right).

The second and more dramatic change to the **Pick SUBGROUP** command is that when a subgroup, Boundary or Interior portion of it is picked in the *Pick Subgroup dialog* the items referenced according to the active editing mode will be highlighted in the *Editor*. This gives a visual feedback of what is being selected rather than the often meaningless selection of named subgroups.

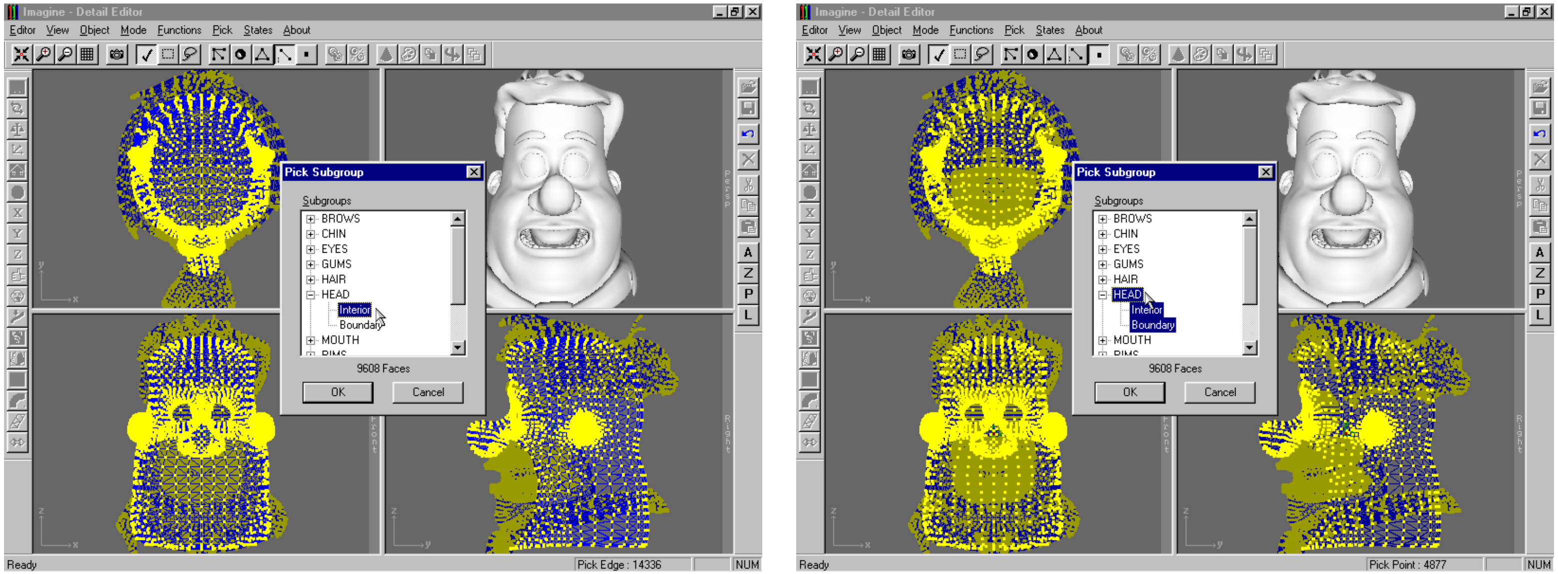


Figure 77: Two further examples showing the Interior edges for the Head subgroup (left) and all the points for the Head subgroup, selected by either clicking the subgroup name or multi-picking both the Interior and Boundary points options (right).