RTrace 1.0

by Antonio Costa

Mac Port by Reid Judd and Greg Ferrar

Introduction

A Brief Description of RTrace

RTrace is a full-featured Ray Tracing program for the Macintosh computer. It can generate both still images and animation sequences.

System Requirements

RTrace will work on any Macintosh running 6.0.4 or later. Its preferred setup is:

- Fast Macintosh (68020 or better)
- Math Coprocessor
- 4 Meg RAM
- 32-bit QuickDraw
- System 7
- QuickTime 1.5

However, it will run on a 68000-based Macintosh without a Math Coprocessor, 32-bit QuickDraw, or System 7, and it can run in as small as a 500K partition.

On a 68000-based Macintosh, or on a Macintosh without a Math Coprocessor, you should use the application called “RTrace/68000.” Otherwise, use the application called “RTrace.” From now on, both programs will be referred to as “RTrace.”

On a Macintosh with only 8-bit Color QuickDraw, RTrace will run, but you will not be able to use the Clipboard to copy images from the Image Window, and the Image Window will not be updated. On a Macintosh with 1-bit (Original) QuickDraw, the image window will not appear at all.

RTrace does not require System 7, but it takes advantage of some of System 7’s features if they are present. It also will take advantage of QuickTime, if it is present. QuickTime is available by anonymous ftp from ftp.apple.com in the /dts/mac/quicktime directory; get the quicktime.hqx file. To play the movies you create with RTrace, you may want a QuickTime movie player; one is available in the same directory; get the movieplayer.hqx file.

A Brief Description of Ray Tracing

Ray Tracing is a process which generates a realistic image from a mathematical description of a scene. It is called “Ray Tracing” because it generates these images by tracing the paths that an actual light ray would take in the scene. By mathematically tracing the paths of thousands of light rays, a ray tracing program is able to create a photographic three-dimensional image of a scene. Since generating a single ray-traced image can take hundreds of thousands of complex mathematical operations, ray-tracing is a slow process. On a very fast Macintosh,

images will be generated fairly quickly. However, if you are using a slower Macintosh (like a Mac Plus), be prepared to wait a long time.

The Fine Print

RTrace is ***absolutely free!!!*** The only requirements are that you not sell it, or distribute it without its documentation. RTrace may not be distributed with any collection of shareware or freeware without the written consent of the authors. However, we'd like to see this program spread as far as possible, so please *do* contact one of us if such distribution is desired.

About the Authors

The original author of RTrace is Antonio Costa, who wrote it for Unix, Vax, PC's, and other systems. The Macintosh port was made by Reid Judd. Greg Ferrar wrote most of the Macintosh user interface, and Reid Judd wrote the rest.

Contacting the Authors

Antonio Costa may be contacted: by email at acc@asterix.inescn.pt.
by "snail mail" at

INESC
Largo Mompilher 22
4100 Porto PORTUGAL

or by telephone at +351+02+321006

Reid Judd may be contacted: by email at ILLUMINATI@AppleLink.Apple.COM.
by "snail mail" at

ILLUMINATI
2617 Sweetbriar Road
Durham, NC 27704

or by telephone at (919)-683-2424

Greg Ferrar may be contacted: by email at gregt@function.mps.ohio-state.edu,
by "snail mail" at

2300 North High Street
Columbus, OH 43202-2902,

or by telephone at (614) 267-8754

Hopes for the Future

RTrace is far from done. We had to stop somewhere, so here's Version 1.0. Here's a sampling of what we have in mind for later versions:

- Support for *real* animation. In other words, it will be possible to have objects in the scene move, rotate, contort — whatever you want. Animation is

currently limited to changing the viewpoint to fly around or through an unmoving scene.

- Converters to/from popular modeling and rendering packages so that RTrace may import scene data files in much the same way Macintosh word processors can import text files.
- Powerful System-7 features. We hope to eventually separate the ray-tracing part from the interface part, so you can run the math-intensive code on a powerful computer while you use the interface on another (networked) computer.
- And much, much more!

All we can do right now is guess what our users will want. If you have suggestion, let us know! RTrace will be an ongoing project. If you're a programmer with access to Think C and would like to donate time and energy to make this program even better, please contact us. And if you have a bug report, *please* let us know!

Getting Later Versions of RTrace

The latest version of RTrace will always be available from asterix.inescn.pt via anonymous ftp, in the /pub/RTrace directory. The Macintosh version will be located in the file(s) in /pub/RTrace/Macintosh/rtrace*.hqx.

A Message from the Author

Here is the README file which is distributed with the unix version of RTrace, written by Antonio Costa. Let's have a big hand for Antonio for writing this ***fabulous*** *ray-tracer and donating it to the world!*

This represents the 7th formal release of the `rtrace' Raytracer. It was written to help me understand how raytracing works, to generate cute images, and generally because I like to program. Feel free to use it for any purpose, I am releasing it into the public domain.

The input format to this ray tracer is called "SFF" or Simple File Format, after using "NFF" or Neutral File Format, which was invented by Eric Haines' for his Standard Procedural Database. The SPD was designed to allow programmers to test their raytracers on databases of varying sizes. While not the end-all to object file formats, it has served me well.

If anyone uses or wants to use NFF, I can send a NFF to SFF converter.

SFF supports the following concepts and primitives:

- point lights
- directional lights
- spot lights with fall-off
- extended lights
- 2 ways of defining surfaces
- spheres
- axis-aligned parallelepipeds
- cylinders
- cones
- bicubic patches
- polygons
- polygonal patches (normals are interpolated from corner points)
- 3D text with high quality
- CSG operations
- 4x4 matrix transformations
- textures
- depth of field
- diffuse distribution
- stereoscopic pair creation

The `rtrace' raytracer supports all of these primitives, with the minor limitation that polygonal patches must be triangles (have only three vertices).

Procedural textures (with 4x4 matrix transformations) are also supported:

- checkerboard
- color blotches
- marble
- bump map
- fbm
- fbm bump map
- wood
- gloss
- image mapping
- waves
- (and many others...)

The output from the raytracer is very simple, and not directly tied to any specific device. It consists of a single line, with format in C style of "%c%c%c%c", which gives the resolution of the image (Width LSByte, Width MSByte, Height LSByte, Height MSByte). It is then followed by Width*Height sets of (red green blue) bytes.

I have lots of filters source code for displaying the ".pic" files, as well as interesting objects that I run across. Filters already exist to display images on Suns, to convert to PostScript, as well as X11 bitmaps for xwud.

I advise you to get a package called the "Fuzzy Bitmap Package" (FBM), that has lots of useful programs for simple image processing, conversion, etc. The author is Michael Mauldin <mlm@nl.cs.cmu.edu>. The Utah Raster Toolkit is also a very good graphics package. Also good is Eric Haines' SPD source code, so you can generate your own fractal spheres, mountains, gears, etc.

Also thanks to the numerous authors whose research into raytracing has seen implementation in this raytracer.

Antonio Costa.

Tutorial

There are two versions of RTrace, the “Power Mac” version (called just RTrace) and the normal version (called RTrace/68000). You should use RTrace if you have a 68020 or better processor, and if you have a math coprocessor. If you have are lacking either of those, you should use RTrace/68000.

To start RTrace, just double-click on its icon in the Finder. RTrace will start up. If you are using a Macintosh which cannot take full advantage of RTrace’s imaging capabilities (one which does not have 32-bit QuickDraw), a message will appear telling you what you’re missing. Click the **Continue** button.

Next, RTrace will show you its nifty About... box. Once you’ve seen enough of it, click on the **Okay!** button. (See the **Preferences** section for information on how to keep that friggin’ box from ever coming up again).

You should see the Log Window now. The Log Window gives you technical feedback about the scene RTrace is processing. Since RTrace isn’t processing anything yet, the Log Window is empty.

Now choose Open... from the File menu. This will bring up a standard Open... window. Use it to open demo.sff, a scene description file provided with RTrace.

RTrace will then present you with a window full of options which looks like this:

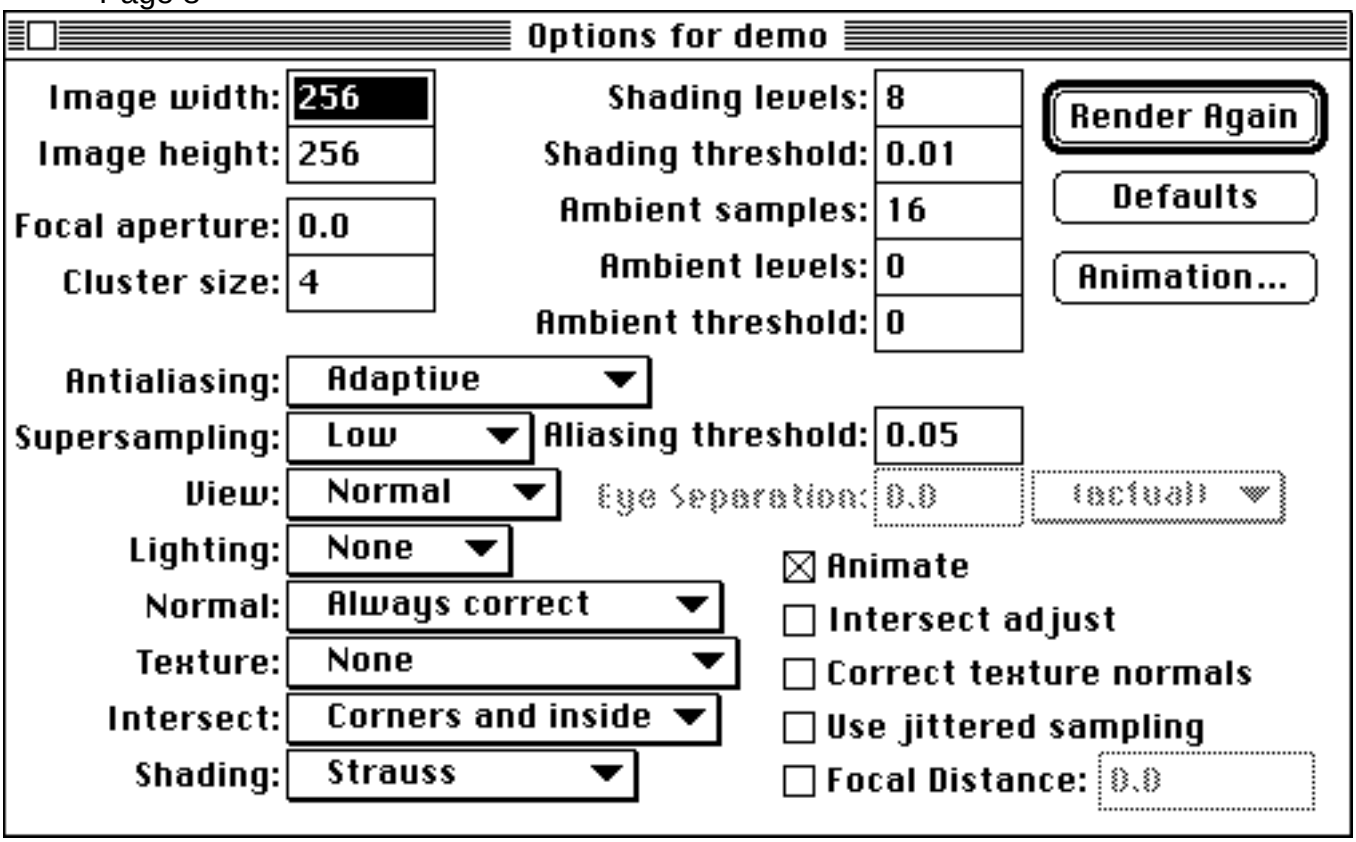

Figure 1: The Options Window

The complexity of the window can be intimidating, but relax. The default options will do nicely for most purposes. As you become more familiar with RTrace, you can experiment with some of the more complex options.

For now, we'll just change the size of the image. The *Image Width* and *Image Height* fields contain the width and height, in pixels, of the image which RTrace will generate. For now, change them both to 100. This will create a 100x100 image, which is small enough that it won't take long even on slower machines.

Now that we've set the options as we would like, we're ready to go. Click on the **Render** button. RTrace will bring up the Status Window and, if you have a color Macintosh, the Image Window. The Status Window gives you visual clues about what RTrace is doing, and provides an indication of how much longer an operation will take. The Image Window displays the image as it is created.

RTrace will first read the scene from disk (from the demo.sff file). Since demo.sff is a very simple scene, that should take almost no time. Then RTrace will begin generating the image, a pixel at a time. This could take a while. If you have a color Macintosh, you will be able to see the image slowly appear in the Image Window.

If not, you will be able to tell how far along it is by looking at the thermometer in the Status Window.

When it's all done, you can save the image by choosing Save... from the File menu. You should save it as a PICT file, since that is the format which most Macintosh graphics applications understand. It will be saved as a 32-bit color PICT file. This file will not be readable on any Macintosh without 32-bit QuickDraw.

Now let's do a simple animation. The **Options Window** should still be visible— click on the Animation button. This brings up the **Animation Window**, which looks like this:

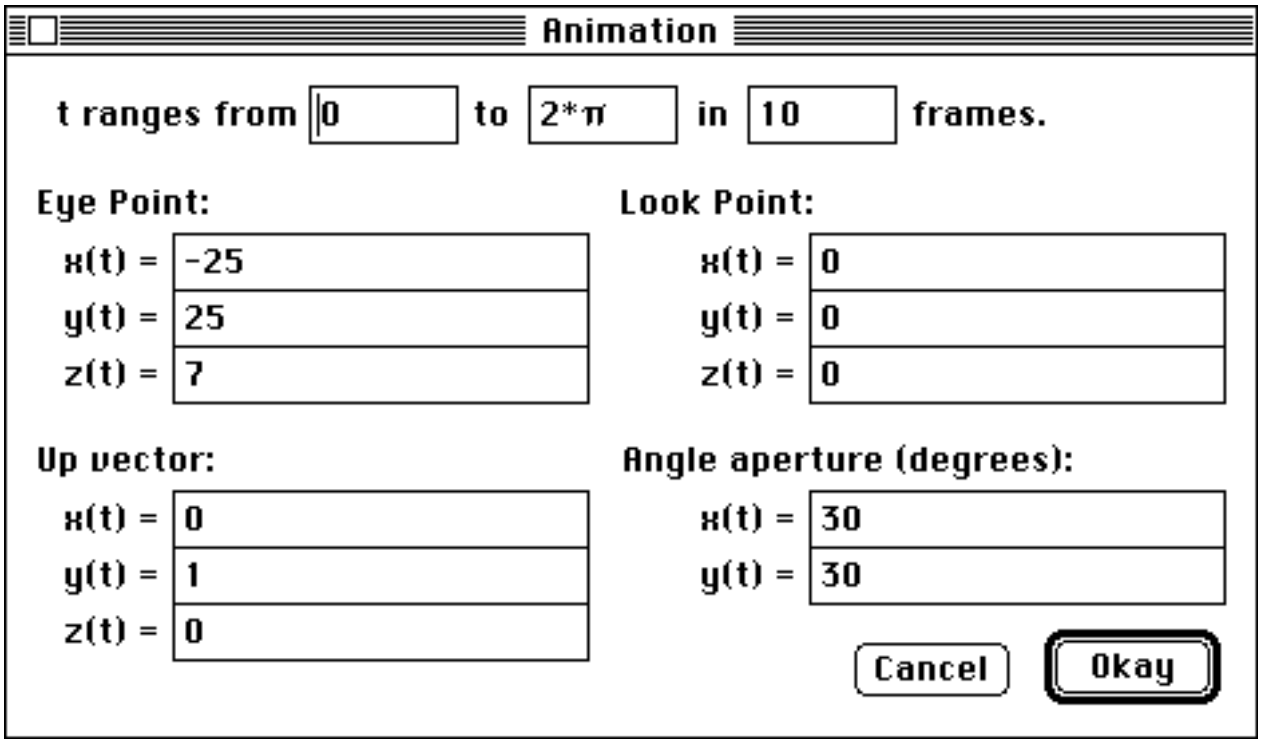

Figure 2: The Animation Window

Again, this can look a little intimidating. Again, the defaults are fine. In a nutshell, the Eye Point is point in space we're looking from. The Look Point is the point we're looking at. The Up vector is a vector which points in the direction which we want to appear to be up. And the Angle aperture is the number of degrees in our view.

Doing animation with RTrace, unlike most other features of RTrace, requires a real understanding of math. You need to know how to define an equation parametrically, and you need to have a good concept of what a vector is. Like the rest of RTrace, it also requires an understanding of space and the traditional

coordinate system. Sorry, that's required. If you don't understand these things, you probably will have a hard time making effective animation.

For this tutorial, we'll lead you through it. What we want to do is make a simple circular orbit around the scene. You can always assume that the default values are a decent viewpoint, so we'll work from there.

Parametrically, a circular path is described by (x, y) = (rsin t, rcos t). Noticing that "up" in demo.sff is in the *y* direction (as indicated by the Up vector (0, 1, 0)), we will plot a course which takes us in a circle around the center of the scene, remaining at a constant height above the ground. A suitable course is (x, y, z) = (-25sin t, 25, 25cos t). We choose 25 for y here because the positive y direction is "up," and we want to be at a constant "height" (above the xz plane). Since the Look Point is (0, 0, 0), we know that the center of the orbit will be directly above the center of the scene.

Now enter the equations. RTrace, like most computer programs, isn't smart enough to read implied multiplication or parentheses. So every time you want multiplication, you need to include an asterisk (*) between the multiplier and the multiplicand. Also, when use a function like sin, be sure to include parentheses around its argument. The correct format for the parametric equations are x(t) = -25*sin(t), y(t) = 25, z(y) = 25*cos(t). Enter these equations now in the Eye Point fields.

The variable t represents time. It is chosen by default to run from 0 to 2*π, which is perfect for a circular orbit. So we will leave the settings for t alone, and leave the number of frames at 10. Click the **Okay** button to return to the Options Window, and then click **Render**.

RTrace will now render ten different views of demo.sff, evenly spaced around the circular path we requested. Actually, they won't all be difference, because RTrace will plot both t=0 and t=2π, which will be identical since sine and cosine are periodic functions with period 2π.

Since RTrace has to render ten frames, this rendering will take ten times longer than the previous one. Be very sure you know what you're doing when you start an animation rendering—it can take a *long* time! If you ever want to abort a rendering, just press ⌘-.

When RTrace is finished rendering, you should save the animation sequence. RTrace can save an animation sequence only as a series of PICT files or as a QuickTime movie. Choose Save Animation... from the File menu, and type a name for the animation sequence, perhaps "demo.anim". If you do not have QuickTime,

use the PICT files option in the popup menu. RTrace will save the ten PICT files in sequence, naming them demo.anim1 through demo.anim10.

If you *do* have QuickTime, select the QuickTime movie option in the popup menu. When you click Save, the QuickTime Compression Window will appear. You may then set several options for compression and frame speed. If you have Millions mode on your monitor, you should try Millions+ for the number of colors. If you have only 256 colors available, try using 256 colors for the number of colors. Choose the other options to be whatever you like, and click Okay. It will be best played in Repeat mode.

Now go back and try playing with some of the options. Try setting the Supersampling to “better” or “best” for better quality images. Try using larger or smaller image dimensions. Try some more animations, with different paths, or with the up vector or the eye point changing with t. Experiment! RTrace is FUN!

The Menus

This section describes the various RTrace pull-menus (in the menu bar) and what they do.

The File Menu

The File menu allows you to open a new scene description, to save an image or animation file, or to quit from RTrace. The menu options are:

Open .sff file...: This prompts you for a scene description file (a text file ending in .sff). Use this when you want to tell RTrace which scene to work with.

Save Image...: This allows you to save the image which RTrace has generated to a file. Use this when RTrace has finished rendering an image, and you want to save it to disk. This gives you the choice of saving the image as a PICT file or as a PPM file. You will usually want to save it as a PICT file, since that is the file format recognized by most Macintosh applications.

Quit: When you choose this option, RTrace aborts any rendering in progress and quits. Use this option when you no longer want to use RTrace.

The Edit Menu

The Edit menu allows you to copy the image to the clipboard, or to change the preferences for RTrace. All other options in the Edit menu are for compatibility with Desk Accessories only. The only options you can use are:

Copy: This option copies the image to the clipboard. This is only available if you have 32-bit QuickDraw. If you do not have 32-bit QuickDraw, you should use the Save Image command to save the image, and then open it from another application.

Preferences...: This option brings up the Preferences Window, and allows you to change the preferences. See **Preferences Window**.

The Windows Menu

The Windows menu lets you show or hide any of the five major windows. When a window is visible, its corresponding menu item will read “Show ... Window.” When it is not visible, the menu item will read “Hide ... Window.”
The available menu options are:

Show/Hide Options Window:

This option will show or hide the **Options Window** . Hiding the **Options Window** during rendering can prevent the windows from cluttering up the screen. It has no impact on performance. See the **Preferences** section for information on how to automatically hide the **Options Window** during each rendering.

Show/Hide Status Window:

This option will show or hide the **Status Window**. Since the **Status Window** provides continual feedback during scene reading and rendering, hiding it can speed thing up significantly. However, it does give valuable information on how much RTrace has done, and how much remains to be done. See the **Preferences** section for information on how to automatically show the **Options Window** during each rendering.

Show/Hide Log Window:

This option will show or hide the **Log Window**. Since the **Log Window** provides some technical information feedback during scene reading and rendering, hiding it can speed thing up a little.

Show/Hide Animation Window:

This option will show or hide the **Animation Window**. Generally, the **Animation Window** will show itself when necessary, and hide itself when it is no longer needed.

Show/Hide Image Window:

This option will show or hide the **Image Window**. Since the **Image Window** is continuously updated during rendering, hiding it can speed thing up a little. Note, however, that if the image is not being kept in memory, the image in the **Image Window** will be incomplete if it is hidden, and then later shown. This does not affect any saved image, which will always be complete. See the **Preferences** section for information on how to automatically show the **Image Window** during each rendering.

The Options Window

The **Options Window** lets you change any of the multiple options RTrace uses to generate the image. The **Options Window** looks like this**:**

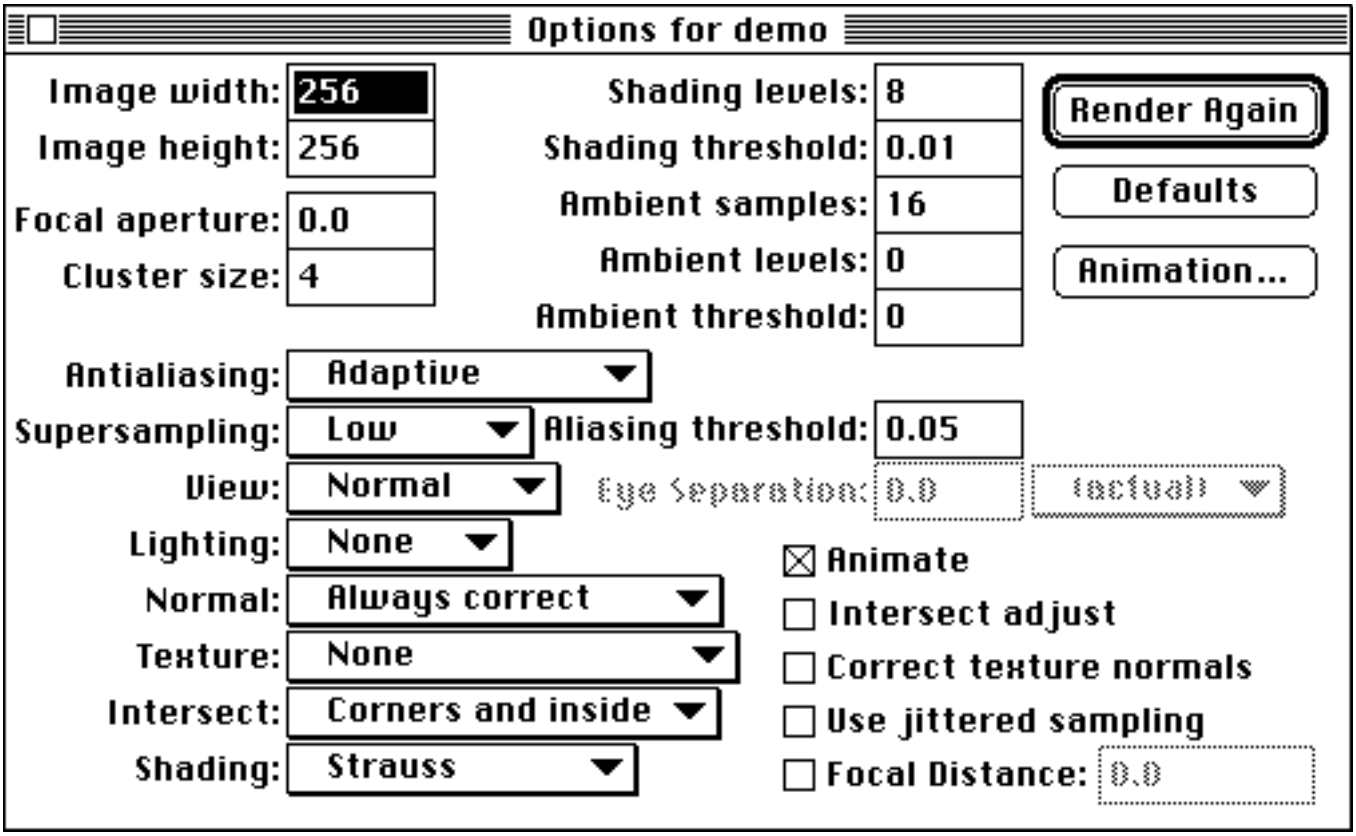

Figure 3: The Options Window

The options (alphabetically) are:

Aliasing Threshold: This is one of the three major controls of image quality, the other two being shading threshold and ambient threshold. It controls pixel supersampling, and ranges from 0 (best) to 1 (poor). The good range is 0.1 to 0.03 (see **Shading Threshold**, **Ambient Threshold**, and **Supersampling**).

Ambient Levels: This option defines the number of shading levels (shading tree depth) in which ambient lighting calculations will be done through ray distribution. Use low values for this option.

Ambient Threshold: This is one of the three major controls of image quality, the other two being aliasing threshold and shading threshold. It controls ambient rays distribution caching, and ranges from 0 (best) to 1 (poor). The good range is 0.01 to 0.00001. A value of 0 means that there is no ambient threshold. (see **Aliasing Threshold** and **Shading Threshold**).

Ambient Samples: This option defines the maximum number of distributed rays to be used in ambient lighting calculations. Use low values for this option.

Animate: When this option is checked, RTrace uses the values in the **Animation Window** to generate multiple scenes. When it is not checked, the values in the **Animation Window** are ignored.

Antialiasing: Antialiasing is a method which smoothes out rough edges caused by visible pixels, eliminating the "jaggies." RTrace can use any of three antialiasing methods: normal supersampling antialiasing, semi-adaptive supersampling antialiasing, or adaptive supersampling antialiasing. Normal supersampling antialiasing should be used with non-zero focal apertures (see **Focal Aperture**, **Supersampling**, **Intersect**).

Cluster Size: As part of RTrace's scene processing, it groups scene objects in clusters. This option lets you specify how many objects there should be in a cluster. Use a low value for sparse scenes, and a high value for dense scenes.

Correct Texture Normals: When this option is checked, RTrace corrects texture normals when textures that modify normals are used, as they may sometimes create strange surface effects. This tends to happen if the scale of the normal perturbation is big.

Eye Separation: This controls the separation between the right and left eyes when using Right eye or Left eye View. It can be either an actual distance, or a percentage of the distance from the Eye point to the Look point (see **View**).

Focal Aperture: This option lets you set the focal aperture of the “camera” which is taking the picture. The default, 0.0, is a pinhole camera. If this is non-zero, there is depth of field, so adaptive supersampling antialiasing will not work so well (see **Antialiasing**).

Focal Distance: This option specifies the focal distance of the “camera” which is taking the picture. If the option is not checked, the distance from the Eye point to the Look point is used.

Image Width: This option lets you select the width of the rendered image, in pixels.

Image Height: This option lets you select the height of the rendered image, in pixels.

Intersect: This option chooses, in adaptive supersampling antialiasing mode, between testing all scene objects or only the objects found at the pixel corners and inside. Testing only at corners and inside greatly reduces CPU time, but with very small objects, is sometimes fails (see **Antialiasing**).

Intersect Adjust: When this option is checked, RTrace avoids some problems with invalid self-intersections. Scenes with text objects should be traced with this option checked.

Lighting: This option controls the generation of shadow rays through non-opaque objects. There may be either no such shadow rays, partial shadow rays, or full shadow rays. If a scene has translucent objects, full or partial shadow rays should be used for the most realistic image.

Normal: Normal is used here in the sense of “perpendicular.” This option lets you control the correction of surface normals, so that it points against the incident ray. With “correct” objects, you should use 1.

Shading: This option chooses between shading models. The options are Normal Phong or Strauss. The Strauss model, developed by Paul Strauss of SGI, is default but slower.

Shading Levels: This option establishes a maximum shading tree depth. When a scene has transparent or reflective objects, it may be important to lower this parameter, or else the tracing never stops. In most cases, there should be no problem allowing it to be large.

Shading Threshold: This is one of the three major controls of image quality, the other two being aliasing threshold and ambient threshold. It controls shading rays propagation, and ranges from 0 (best) to 1 (poor). The good range is 0.01 to 0.001 (see **Aliasing Threshold** and **Ambient Threshold**).

Supersampling: Supersampling refers to the antialiasing process where a single pixel is sampled many times, at slightly different positions inside the pixel. A greater number of samples results in smoother edges. However, supersampling significantly slows the rendering process. The choices are none, low, medium, and high. Choosing higher supersampling values improves the image quality, but slows rendering. “Medium” is a good choice for high resolutions, but “High” gives the highest image quality (and takes the longest time). (See **Antialiasing**, **Aliasing Threshold**).

Texture: This option lets you specify where the texture information appears in the scene description (.sff) file. There may be either no textures, or textures defined inside the objects field, or textures defined after the objects field. If you try to read a file with the wrong texture format, RTrace will generate an error. In that case, just change this option and try again. Currently, we only support .PPM format for image textures so you will need to convert PICT files to .PPM in order to use them in image texture mapping.

Use Jittered Sampling: When this option is checked, RTrace uses jittered sampling. Sometimes checking this produces better images from scenes with small tricky details.

View: RTrace supports stereoscopic viewing, where two images are rendered from slightly different viewpoints, to emulate the separation of human eyes. This option lets you choose to render from the Left eye's view point or from the Right eye's viewpoint. If the view is Normal, no stereoscopic offset takes place (see **Eye Separation**).

There are also three buttons in the Options dialog:

Render: This begins a new rendering, using the current options on the most recently opened scene description (.sff) file.

Defaults: This resets all options in the Options Window to their default settings.

Animation: This brings up the Animation Window.

The Animation Window

The **Animation Window** lets you define parametrically the Eye point, the Look point, the Up vector, and the Angle aperture as a function of time, t. The **Animation Window** looks like this:

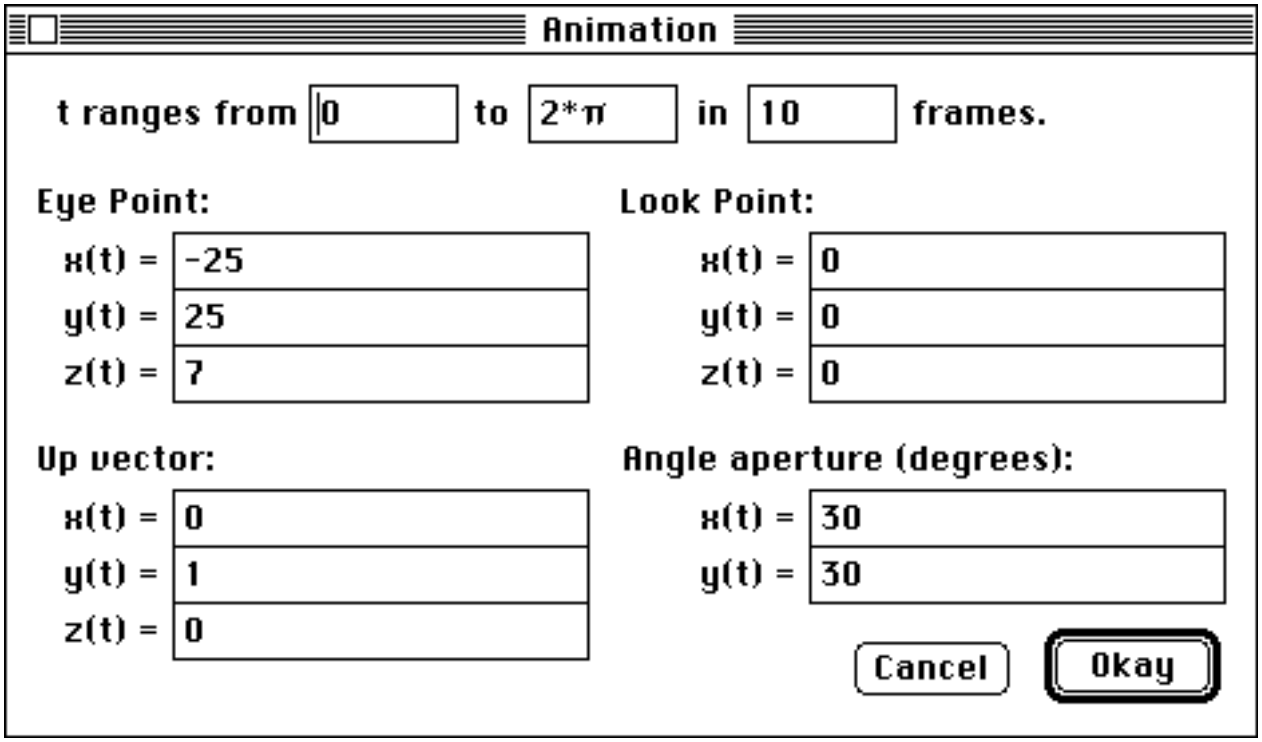

Figure 4: The Animation Window

The options are:

t range: These three fields at the top of the window completely describe the behavior of the variable t. During rendering, t will range from the left value to the middle value in the number of steps specified at the right. The steps will be evenly spaced between the max and the min, and both the max and the min will be included as endpoints. The beginning and end values may be mathematical expressions, but they may not contain the variable t. The number of frames must be an integer greater than zero.

Eye Point: This allows you to select the point from which the scene is viewed. The three fields specify the x, y, and z coordinates of the Eye point, and may be mathematical expressions in t.

Look Point: This allows you to select the point towards which we are looking. The three fields specify the x, y, and z coordinates of the Look point, and may be mathematical expressions in t.

Up Vector: This allows you to select the direction which is considered “up”. This direction will point towards the top of the screen in the rendered image. The three fields specify the x, y, and z components of the Up Vector, and may be mathematical expressions in t.

Angle Aperture: This allows you to select the angular aperture of the view, specified in *degrees*. For instance, angles of 90° means that the entire hemisphere will be displayed, while an angle of 45° means only a pyramidal portion, with 90° angles at the vertex (the Eye Point) will be displayed (see diagram). The two fields specify the horizontal and vertical Angle Apertures, and may be mathematical expressions in t.

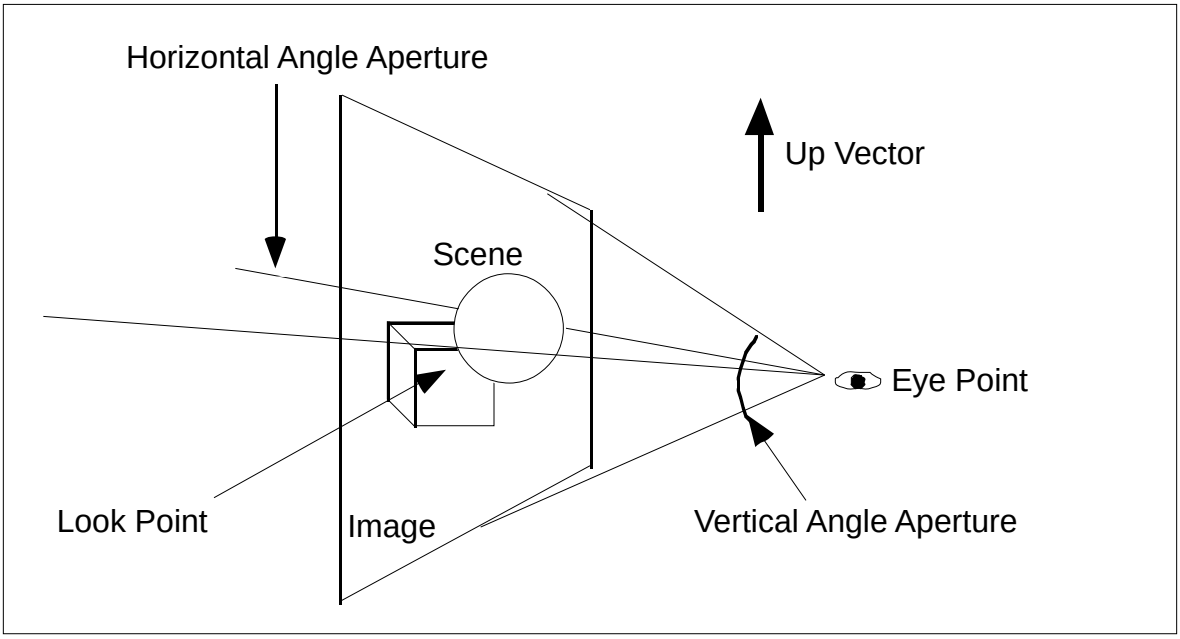

Figure 5: The Animation Parameters

The Status Window

The Status Window gives you feedback on what RTrace is doing at a given time. The status window looks like this:

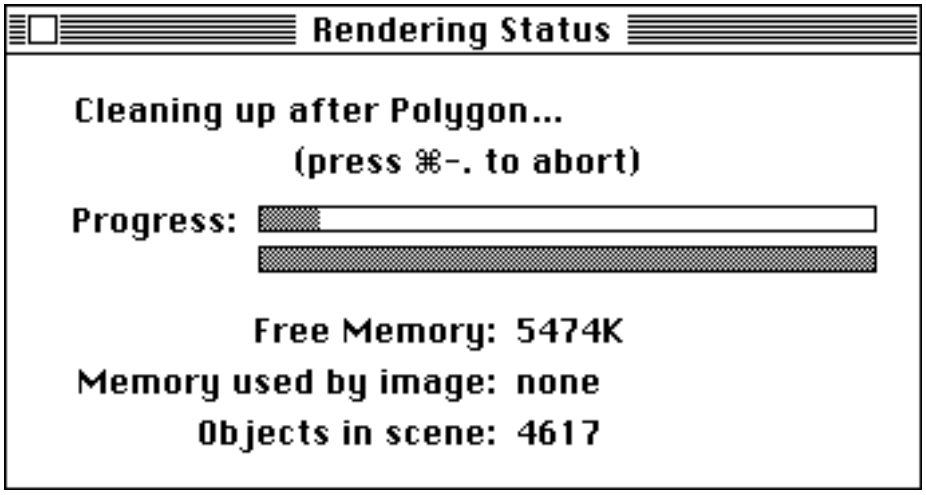

Figure 6: The Status Window

The top line is a text description of what RTrace is doing at the moment. Below that there are two thermometers. The upper one indicates the amount of time left in the current major task, which the bottom one shows the amount left in a minor task. Major tasks include reading the scene, enclosing the scene, and rendering. Minor tasks include reading a single polygon from the scene.

The next line down shows the amount of free memory in RTrace's MultiFinder partition. If you are using Finder, it shows the total free memory. The number will decrease as the scene is read, and as memory is allocated for other purposes.

Beneath that is the amount of memory currently used by the image, if it is being stored in memory. In the case above, the image is not being stored in memory, so no memory is used.

Finally, the number at the bottom indicates the number of objects in the current scene. Larger scenes with more objects require more memory to render.

The Log Window

The Log Window is a scrollable window which contains technical output which RTrace generates during the scene reading, enclosing and rendering. It looks like this:

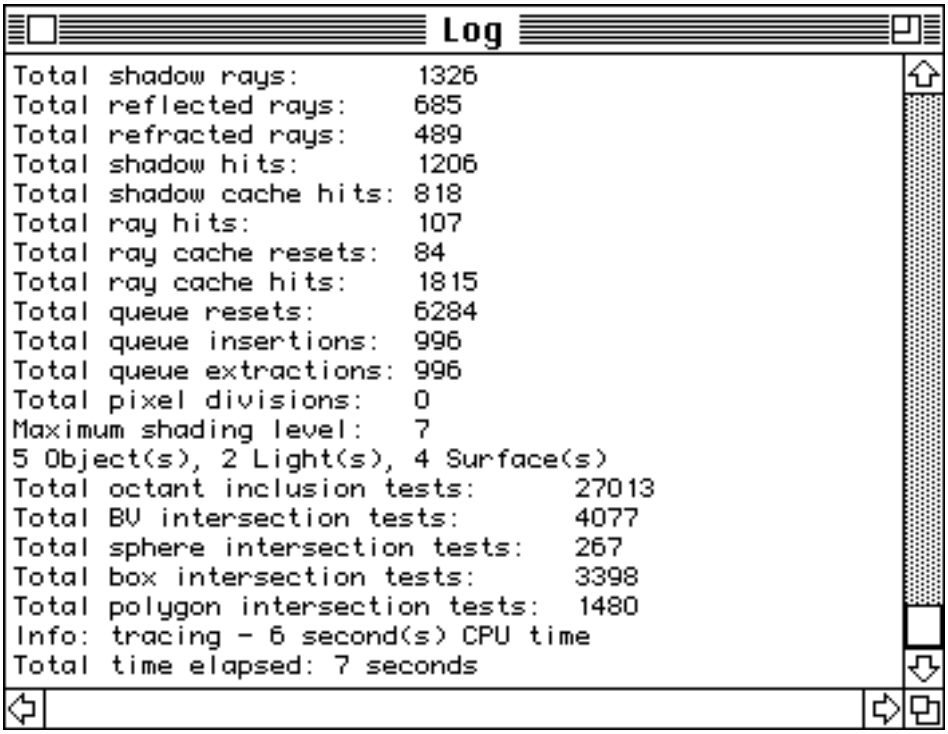

Figure 7: The Log Window

The log window is limited to 1000 lines of text. After this limit is exceeded, the oldest lines will disappear from the top. If you want to see a complete, unlimited log of the rendering, look at RTrace Log File, which RTrace creates in its directory each time it is launched.

The Image Window

The Image window is a standard window which contains the image being rendered. It looks like this:

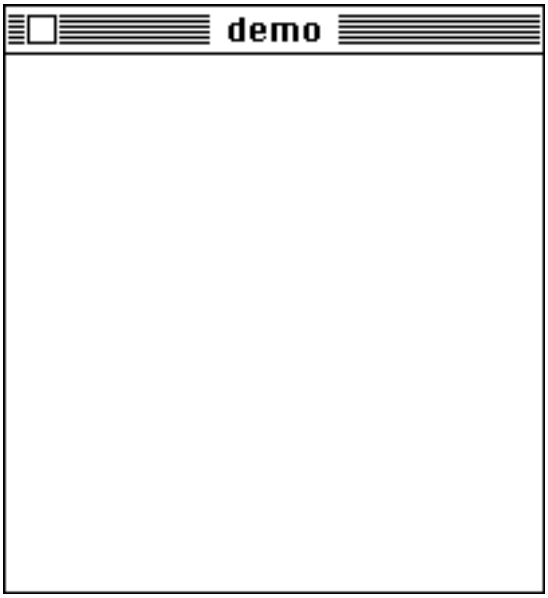

Figure 8: The Image Window

The size of the Image Window will vary depending on the size of the image. While the image is rendering, the Image Window will contain a partial image.

If the Keep Image In Memory option is checked in the Preferences Window, a copy of the image will be kept in memory. If part or all of the Image Window is obscured by another window, and then revealed again, it will be redrawn from the copy in memory. This is expected in a Macintosh application. However, due to the large number of colors in the image, it can take a huge amount of memory to keep the image in memory. In low memory situations, therefore, it is possible to *not* keep the image in memory, by checking off the option in the Preferences Window.
Note, however, that in this case the image window will not be properly redrawn if it is obscured. If the image is not being kept in memory, you should try to avoid obscuring the Image Window.

On Macintosh computers without 32-bit QuickDraw, it is not possible to keep the image in memory. In such a case, the Keep Image in Memory option is always off.

On Macintosh computers without Color QuickDraw, there is no Image Window, and the image cannot be seen while it is rendering.

The Preferences Window

The Preferences Window is brought up by selecting Preferences… from the Edit Menu. It looks like this:

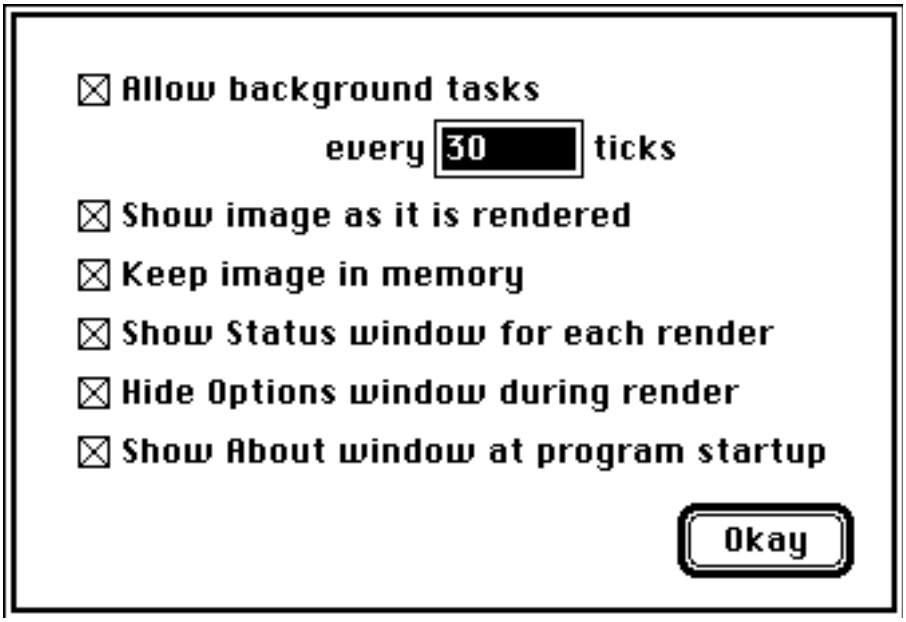

Figure 9: The Preferences Window

The Preferences Window allows you to set various RTrace options which will carry over from session to session. The options are:

Allow background tasks:

When this option is checked, RTrace will act like a “nice” Macintosh application, and will give some of its time to any programs running in the background. The exact amount of time it will give depends on the value of the number below, which is in 1/60 second intervals. In the above case, RTrace will stop every 30 intervals (every half second) to give up a little time to background programs, and to check for any events (like abort). Disabling it will cause RTrace to run “blind”; it will not stop for anything until it is done with it’s task. When this option is disabled, RTrace runs significantly faster. However, it is not possible to do *anything* else when RTrace is in this mode, and it is not possible to abort a rendering.

Show image as it is rendered:

When this option is checked on, the Image Window will be made visible each rendering, and the image will be drawn in the Image Window as it is rendered.

Keep image in memory:

When this option is checked on, the image being rendered will be stored in memory. This allows the Image Window to be updated when it becomes obscured and then is revealed again. However, it can take a lot of memory to store the image in memory (actual number of bytes = image height x image width x 3). In a low-memory situation, try turning this option off. The memory will be made available for scene data and other necessary information. However, the Image Window will not be updated if it is obscured and then revealed. When this option is off, you should try not to obscure the Image Window if you want the image on the screen to remain intact. Note, however, that any saved image will be intact, regardless of any settings.

Show Status window for each rendering:

When this option is checked on, the Status Window is shown each time a rendering begins. This will speed things up a bit, since drawing all the status information takes time.

Hide Options window during each rendering:

When this option is checked on, the Options Window is hidden each time a rendering begins. This may help to prevent clutter, since the Options Window is usually not needed while an image is rendering.

Show About window at program startup:

When this option is checked on, the About Window is shown each time RTrace is launched.