Chapter 1 Getting Started

Chapter Overview

Initiating NCSA Isosurface Visualizer

Testing ISOVIS

Exiting ISOVIS

Manipulating ISOVIS

Command Line Parameters

Chapter Overview

This chapter offers a tutorial which introduces and describes NCSA Isosurface Visualizer. In addition, all of the program's command line parameters are explained in detail. Examples are included throughout to demonstrate how the Isosurface Visualizer, or *ISOVIS*, can be used and also its ease of use.

This program assumes that you are able to work with HDF (*Hierarchical Data Format*) files and the UNIX system. If you are unfamiliar with either system or need more detailed information regarding certain procedures, please refer to their respective user's manuals. (Refer to Appendix A, "Obtaining NCSA HDF and Other Software," for information on obtaining NCSA HDF.)

Initiating NCSA Isosurface Visualizer

Testing ISOVIS

Isosurface Visualizer is a 3D graphics visualization tool written for the UNIX environment. It is a non-interactive command-line driven program which is executed like most UNIX commands. To begin this tutorial, you must first make sure Isosurface Visualizer is in your path.

1. Type '**isovis**' to verify that the program is present. If it is, you should see the following displayed on your screen:

```
+-----------------------------+
|  isovis, version: x.xx  date: month, year        [P] |
|  public domain software, NCSA, Urbana-Champaign, IL  |
+-----------------------------+

Usage  : isovis [-options] <3d_sds.hdf> <threshold>
```

```
-bc <br bg bb>
-d <0|1>
-k <Ka Kd Ks>
-lc <lr lg lb>
-ll <lx ly lz>
-ns <Ns>
-ntsc (use ntsc size viewport)
-o <hdf.r24>
-r <rx ry rz>
-s <sx sy sz>
-t <tx ty tz>
-v <xorg yorg xsize ysize>
-p (print current settings)
-vset <hdf.vset>
-obj <obj.file>
```

NOTE: See the section of this chapter entitled, "Command Line Parameters," for a thorough definition of the capabilities and extents of the above parameters. To test ISOVIS and manipulate its windows:

2. Execute ISOVIS without any options. You will need a three-dimensional (3D) Scientific DataSet (SDS) file in HDF format. Type the following (*filename* represents any valid 3D HDF SDS file; *threshold* represents any valid scalar value contained within the dataset):

   isovis filename *threshold*

   If the file is present, a small outline of a box appears at the cursor.

3. Hold down the left mouse button and drag to open and enlarge the window. Once you release the mouse button, you can no longer manipulate the size of the window.

4. To move the box around the screen, hold down the middle mouse button and position the pointer in the top bar area of the window. The arrow will turn into a circle. Click and drag to a desired area.

At this point, an image resembling your data should appear in the window. The larger your 3D input data, the longer the image takes to render.

Exiting ISOVIS

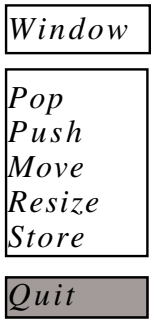
To exit ISOVIS:

1. Position the cursor in the graphics window and press the [Esc] key on the keyboard.

or

2. Hold down the right mouse button. The pop-up window, shown in figure 1.1, appears. Choose "Quit" to leave the graphics window. If you do not position the arrow in the window while you are either typing in or trying to exit it, the results will be unpredictable.

Figure 1.1 Pop-up Window



Manipulating ISOVIS

User-Defined View

To observe the same data from an alternate view which you specify, change the command line parameters. Although you may simultaneously alter as many parameters as are available, for the purpose of this tutorial, alter only the rotation parameter and the scale parameter. (See the section in this chapter entitled, "Command Line Parameters" for more information on these two variables.)

1. Enter: **`isovis -r 50.0 190.0 0.0 -s 6.0 4.0 -0.5\ `*`filename threshold`*

   A small outline of a box appears at the cursor.

2. Enlarge the window by holding the center mouse button down and dragging from the right-hand corner of the box. A rendered image of your data appears. Notice that your image has rotated 50 degrees to the left (counterclockwise) along the x axis and 190 degrees to the left along the y axis, but has not rotated at all along the z axis. The image has also changed in scale; it is now six times as large along the x axis, four times as large along the y axis, but half as large along the z axis as compared to the default ISOVIS image. You have successfully altered the orientation of geometry in your image.

3. Exit ISOVIS.

The above examples demonstrate how easy it is to create a 3D visualization of your data using NCSA Isosurface Visualizer.

When working with many command line parameters simultaneously, you may want to create a file containing all the parameters by using your favorite editor. Doing so enables you to easily adjust the parameter dimensions without repeatedly entering each individual command line.

Command Line Parameters

The following section describes each of the command line parameters in detail, beginning with a description of the default parameters.

isovis: no options

You may execute ISOVIS without any optional command line parameters.

The parameters default internally to the following:

```
-r 0.0 0.0 0.0
-t 0.0 0.0 0.0
-s 1.0 1.0 1.0
-ll 50.0 50.0 50.0
-lc 0.9 0.9 0.9
-k 0.3 0.7 0.0
-ns 3.0
-bc 0 0 0
-d 1
```

In effect, if you specify these options on the command line with the parameters above, the same rendered image results. The meaning of these parameters follows.

-bc <br bg bb>: background color

The *-bc parameter* specifies the color of the image's background. Values are given as floating-point numbers with a valid range of [0.0...1.0]. *br bg bb* specify the red, green, and blue intensity of the image background, respectively.

-d <0|1>: display image flag

The *-d parameter* specifies whether the image should be displayed or not. A value of 1 displays the image (default), while a value of 0 doesn't. This option is useful if you want to save the polygons to a file and don't want to spend any time rendering the polygons on the screen.

-k <Ka Kd Ks>: coefficients of object material properties

The *-k parameter* specifies the percentages of ambient, diffuse, and specular material properties to be used in determining the ambient, diffuse, and specular RGB coefficients derived from the absolute RGB color at a given polygon or vertex. For a more detailed description of these values, see Chapter 3, "Technical Notes." Values are given in floating-point with a valid range of [0.0...1.0]. Ideally, the sum of Ka + Kd + Ks <= 1.0. The sum may exceed 1.0 to accentuate one or more of the parameters in the rendered image.

-lc <lr lg lb>: light source color

The *-lc parameter* specifies the red, green, and blue color of the light source, respectively. Values are given in floating-point with a valid range of [0.0...1.0]. You may enter values greater than 1.0 to saturate a given color intensity over the surface of an object.

-ll <lx ly lz>: light source location

The *-ll parameter* specifies the absolute light source location in a right-handed Cartesian coordinate system directed towards the world coordinate origin. The light source currently emits parallel rays of light which do not attenuate in intensity over distance. Therefore, the distance the light source is from the object along an arbitrary vector does not affect the

intensity of light which reaches the object. Direction of the light source is relevant and thus can only illuminate the side “visible” to the light source from a given location. The light source arguments are floating-point where any valid world-coordinate location is acceptable.

-ns <Ns>: specular reflection exponent

The *-ns parameter* specifies the size of the specular reflection when modeling shiny surfaces. *-ns* only takes effect when the Ks argument of the -k parameter is greater than zero. Ns is a floating-point value with a valid range of greater than 1.0. Values close to 1.0 model less shiny surfaces with large specular high-lights, while values close to 200.0 model very shiny surfaces with small focused specular high-lights. Typically, values greater than 200.0 are not effective. The specularly reflected light appears on the object surface in the color of the light source. To eliminate specular reflection entirely, use zero for the Ks argument of the -k parameter.

-ntsc (use ntsc size viewport)

The *-ntsc parameter* creates an *ntsc size viewport.* Current ntsc resolution is 640 in X by 480 in Y. The viewport’s origin is positioned at the origin in screen coordinates, i.e. the lower left of the workstation screen. There are no arguments for this parameter.

-o <output_file_name>: HDF raster output file name

The *-o parameter* specifies the name of the HDF raster output file which will store the rendered screen image. Normally, ISOVIS does not exit by itself when an object has been rendered. However, when the -o option is in effect and the object has been rendered to the screen, the image is read from the screen and written to disk as a 24-bit HDF raster image file, and ISOVIS is terminated. This option is only available when the -d parameter is 1, meaning that the image was displayed.

-r <rx ry rz>: rotation

The *-r parameter* specifies rotation about the object’s coordinate system in Cartesian coordinates. The world coordinate system is a right-handed Cartesian coordinate system indicating that the positive x axis is to the right, the positive y axis is up, and the positive z axis is towards the viewer. Rotation angles are specified in degrees. A positive rotation angle about a given axis causes the object to rotate counter-clockwise. Conversely, a negative rotation angle about a given axis causes the object to rotate clockwise. This movement is apparent when you view the object from a positive point on that same axis while looking towards the origin. The rotation order is x followed by y followed by z. The rotation arguments are floating-point with typical values ranging from [–360.0...360.0].

-s <sx sy sz>: scale

The *-s parameter* specifies the scaling of the object in each of the x, y, z axes. -s 1.0 1.0 1.0 preserves the object’s original geometry without distorting it, while a scaling of -s 0.0 0.0 0.0 causes an error. For example, -s 2.0 2.0 0.25 scales the object twice as big in x and y, and one-fourth as

high in z. The scaling arguments are floating-point.

-t <tx ty tz>: translation

The -t parameter specifies the amount of absolute translation (not relative) of the object from the world coordinate origin in a right-handed Cartesian coordinate system. By default, the origin is in the center of the window. Values can be any valid floating-point numbers, although you'll typically want to keep them in the range [0.0...maxdim] where *maxdim* is the largest dimension of your dataset.

-v <xorg yorg xsize ysize>: viewport position and size

The *-v parameter* specifies the viewport's screen position and its size. *xorg* and *yorg* specify the position of the viewport's origin in screen coordinates. Note that the screen origin is in the lower left-hand corner. Typical values for xorg and yorg are in the range [0...1279] and [0...1023], respectively. *xsize* and *ysize* specify the viewport size in pixels. Typical values for xsize and ysize are in the range [40...1280] and [40...1024], respectively. Values outside of these ranges are permitted, but cause a portion or the entire viewport to be clipped to screen boundaries. Viewport arguments are integers.

-p (print current settings)

The *-p parameter* gives the user feedback on the values being used for all command line parameters at the time of execution. These values are printed to the standard output device. There are no arguments for this parameter.

-vset <output_file_name>: HDF Vset output file name

The *-vset parameter* specifies the name of the HDF Vset output file which stores the polygons created by ISOVIS. The Vset can then be used by other software, such as NCSA PolyView.

-obj <output_file_name>: object output file name

The *-obj parameter* specifies the name of the object output file which stores the polygons created by ISOVIS. The object file is a simple ASCII text file that can be easily converted into other formats, or it can be used by other software, such as Personal Visualizer by Wavefront Technologies, Inc.