Chapter 1 Getting Started

Chapter Overview

Initiating NCSA Height-Color Visualizer

Testing HCVIS

Exiting HCVIS

Manipulating HCVIS

Command Line Parameters

Chapter Overview

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

This program assumes that you are able to work with HDF 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.

Initiating NCSA Height-Color Visualizer

Testing HCVIS

Height-Color 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 Height-Color Visualizer is in your path.

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

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

Usage  : hcvis [-options] <height_file> <color_file>

  -r <rx ry rz>
```

```
-t <tx ty tz>
-s <sx sy sz>
-ll <lx ly lz>
-lc <lr lg lb>
-k <Ka Kd Ks>
-ns <Ns>
-ad (use alternate defaults)
-ntsc (use ntsc size viewport)
-v <xorg yorg xsize ysize>
-i <illumination_model[0,1]>
-c <color_palette_file_name>
-bc <br bg bb>
-axes <x_string y_string z_string>
-O8 (use 8 bit/pixel output)
-R (use RELIEF representation)
-W (x_sample y_sample line_width>
-p (print current settings)
-o <output_file_name>
```

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 HCVIS and manipulate its windows:

2. Execute HCVIS without any options. You will need an image file in HDF format or a 2D scientific data set in HDF format. Type the following (filename represents any valid 2D HDF raster or scientific data file):

   hcvis *filename filename*

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

3. Hold down the left-most 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 desired area.

At this point, an image resembling your data should appear in the window. The larger your 2D input data, the longer the image takes to render.

Exiting HCVIS

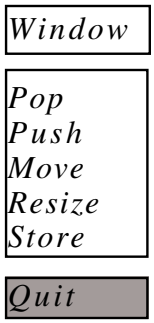
To exit HCVIS:

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

or

2. Hold down the far-right button on your mouse. 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 you are either typing in or trying to exit, the results will be unpredictable.

Figure 1.1 Pop-up Window



Manipulating HCVIS

Programmed View

To observe the same data from an alternate view established within the program:

1. Enter: **`hcvis -ad filename filename`**

2. Open the window as before and observe the rendered image.

3. Leave the window by following the steps in the section, “Exiting HCVIS.”

NOTE: Each time you want to see an alternate view of your image, you should first exit the current image of your 2D scientific data set or image file by following the steps outlined in the section, “Exiting HCVIS.” Failure to do so will create another window consuming additional available memory.

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: **`hcvis -r 50.0, 190.0, 0.0 -s 6.0, 4.0, -0.5\ filename filename`**

   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 HCVIS image. You have successfully altered the orientation of geometry in your image.

3. Exit HCVIS.

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

hcvis: no options
You may execute HCVIS 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
-lx 0.0 0.0 1.0
-lc 1.0 1.0 1.0
-k 0.3 0.7 0.0
-ns 3.0
-i 1
-bc 0 0 0
```

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.

-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].

-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. Since geometry is normalized by HCVIS to fit in a window with extents from [–1.0...1.0] along each axis, typical translation values will be floating-

point values also in the range [–1.0...1.0].

-s <sx sy sz>: scale

The -s parameter specifies the scaling of the object in each of the XYZ 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. -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.

-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.

-lc <lr lg lb>: light source color

The -lc parameter specifies the Red, Green, 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.

-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 (depending on the illumination model: see the -i parameter description). 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.

-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.

-ad (use alternate defaults)

The -ad parameter offers a convenient set of alternate defaults which produces a more useful orientation and representation of the data than

the original defaults. No arguments exist for this parameter. Use the -p parameter to view the values used for the alternative defaults. The -ad parameter overides the following parameters if they are present on the command line: -r -t -s -ll -lc
-k -ns.

-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.

-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, resulting in a portion or the entire viewport being clipped to screen boundries. Viewport arguments are integers.

-i <illumination_model>: illumination model

The -i parameter specifies which illumination model to use when rendering the image. Currently, the only supported illumination models are 0 and 1.

• **I**llumination Model Zero
  Illumination model zero eliminates all lighting from the scene while preserving the original color information.

•Illumination Model One
  Illumination model one uses a simple lighting model based on Lambert's cosine law. The model adjusts color values at each polygon vertex proportional to the cosine of the angle between the polygon normal and the light source vector. This model tends to high-light gradations and enhance dimensionality of the 3D object.

More advanced lighting models based on the vertex normals may be supported in future releases. See Chapter 3, "Technical Notes," for a more detailed discussion of available illumination models. The illumination model argument is integer.

-c <color_palette_file_name>: color palette file name

The -c parameter specifies the name of a color palette file which is used to color the <color_file> named in the command line. This option is only relevant for 8-bit pseudo colored images or scientific data sets and does not have any meaning for 24-bit true colored images. Since the 8-bit raster image set (RIS8) in HDF supports the storing of a palette within the image file, this command may not be necessary. However, the -c command overides any and all other color palettes encountered. Internally, the reader for the program tries first to read the palette file in HDF

RIS8 format. If this fails, it reads the file as raw binary using the first 256 bytes for red, the next 256 bytes for green, and the next 256 bytes for blue. A warning message appears on your screen if the palette file is read as raw binary. The raw binary format is included for convenience and probably will not be supported in future releases.

-bc <br bg bb>: background color

The -bc parameter specifies the color of the background used in the image. Values are given as integers with a valid range of [0...255]. Br bg bb specify the red, green, blue intensity of the image background, respectively. When using the -bc parameter with the -O8 parameter, the current color map is modified such that entry zero is set to the background color. A warning message appears to this effect. Therefore, if there are zeros in your data and they are colored with the current color map, these values are alternately colored in the background color when the -O8 and -i 0 options are in effect. The color map is not modified if the -O8 parameter is not in effect.

-axes <x_string y_string z_string>: draw axes with axes labels

The -axes parameter draws the X-Y-Z axes in a right-handed cartesian coordinate system such that these axes are aligned with the positive X,Y, and Z axes, respectively. Values are character strings drawn at the end of the respective axes. In color map mode (i.e. -O8 and -i 0), the axis will be drawn in the available color map entry most complementary to the background color. In true color mode, the axes will be drawn in the color exactly complementary to the background color. Multiple words and spaces can be placed in a single string value by enclosing the desired string in double quotes, e.g. -axes “pressure range” “temperature range” ”magnitude”.

-O8 (use 8 bit/pixel output)

The -O8 option uses 8-bits per pixel if illumination model zero (i.e., -i 0) is active. If the -o option is in effect, -O8 will do a 24 to 8-bit color compression of the image using a sophisticated algorithm before writing to the file. The 8-bit compressed image will be written to the file without being displayed on the screen. Normally, all rendering defaults to 24-bits per pixel true color. However, with illumination model zero (see the -i parameter description), it is possible to render the image using an 8-bit per pixel pseudo color representation instead of the 24-bit per pixel true color representation. The differences should be understood. See Chapter 3, “Technical Notes,” for a more detailed discussion of this topic. There are no arguments for this parameter.

-R (use RELIEF representation)

The -R parameter specifies that the object should be rendered as a relief representation of the object. Using the relief representation, the object’s height geometry Z scale (i.e. height off the XY plane) is set to zero. Gradations in data values in the color field are high-lighted using a lighting model. The -R option forces the -i illumination model to be non-zero indicating that a lighting model is necessary for the relief representation. To high-light gradations without using any coloration from the color field file, substitute the word “WHITE” on the command line instead

of the color field file. For example, type:

```
hcvis hfile.hdf WHITE
```

rather than,

```
hcvis hfile.hdf cfile.hdf
```

where hfile.hdf is the height field file and cfile.hdf is the color field file.

This assigns a full white material color to all of the height field nodes which can then be used by the active lighting model to high-light data gradations only. If the -R and -W are both given on the command line, the -R parameter overides the -W parameter and
-W is ignored. There are no arguments for this parameter.

-W <x_sample y_sample line_width>: use WIREFRAME representation

The -W parameter specifies the object should be represented by lines. The arguments x_sample and y_sample specify the sample rate at which to retrieve data values from the 2D height field. Valid values for x_sample and y_sample are [1...height_xsize] and [1...height_ysize], respectively. The line_width argument specifies the line width in pixels used when drawing the wireframe object. Typical values for line width are [1...5]. The wireframe representation is drawn much faster than the other representations and is useful for setting up object orientation parameter arguments (i.e. -r -t -s). Overlapping lines are drawn correctly. Wireframe representation 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.

-o <output_file_name>: HDF raster output file name

The -o parameter specifies the name of the HDF raster output file which stores the rendered screen image into. Normally, HCVIS does not exit by itself when the 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 an 8-bit or 24-bit HDF raster image file and HCVIS is terminated. The default output is 24-bit. However, if the -O8 option is in effect the image is written in 8-bit RIS8 HDF format. If the image is rendered as 24-bit and the -O8 option is in effect, the 24-bit image will be compressed to 8-bits using a sophisticated color quantitization algorithm before writing to disk.