LONI SHAPEVIEWER

(version 1.1.12)

- OVERVIEW
- INSTALLING AND COMPILING
- SYSTEM REQUIREMENTS
- LAUNCHING
- MENUS AND BUTTONS
- DISPLAYS
- COMMAND LINE ARGUMENTS
- SCENES
- DRAWING MODES
- READING SHAPES
- MESSAGE PANE
- COMMAND LINE OPTIONS
- APPENDIX 1 - UCF FILES
- APPENDIX 2 - MULTIGRID UCF FILES
- APPENDIX 3 - OS X TERMINAL
- APPENDIX 4 - Coloring Data with ColorMaps

OVERVIEW

Neuroimaging Researchers often need to see geometrical shapes relevant to their research. These shapes are typically the surfaces of interesting brain structures. The visualization tools now available do not support the variety of file formats in use, are often far more complex than what is warranted by the needs of the researchers, and are seldom portable across different computers and operating systems.

INSTALLING AND COMPILING

Compilation:

Currently, only the executable is distributed via the website. Compilation is therfore not required.

Installation:

Before running the ShapeViewer, you must verify that you have both Java version 1.4 (or later) and Java3D installed on your machine. You may install the ShapeViewer in any directory. To start the ShapeViewer, you may open the ShapeViewer Jar file directly (double-click), or use a shell script or batch file (Windows) appropriate for your operating system.

SYSTEM REQUIREMENTS

Java

ShapeViewer uses the Java bytecode interpreter and it's presence is required on your computer to run ShapeViewer. If you are not sure whether Java is installed , please start a command shell (see Appendix for help with this) and type the following:

```
java -version
```

You should seee a response similiar to the following:

```
Java(TM) 2 Runtime Environment, Standard Edition (build 1.4.2_05-141.4)
Java HotSpot(TM) Client VM (build 1.4.2-38, mixed mode)
```

ShapeViewer requres that java version 1.4.1 or greater be installed. If you do not have a compatible version of java installed please contact your system adminstrator

for help installing Java.

Java3D

ShapeViewer relies on the three dimensional graphics display capabilities of the Java3D extension and cannot run without it. If you launch ShapeViewer and do not have Java3D installed on your computer you will see a message similar to the follwing displayed.

Java3D: not found

Please forward the message to your system administrator for help with installing Java3D on your computer;

LAUNCHING

To start the ShapeViewer program start a Java interpreter and use the "-jar" option to refer to the ShapeViewer JAR file.

Operating System	Example command line	Notes
Macintosh OS X	java -jar ShapeViewer.jar [optional command line arguments]	Type this in the "Terminal" comand window. See Appendix for more information.
MS Windows	runShapeViewer.bat	Windows requires a batch script.
Linux, Solaris, or other similar Unix computer	java -jar ShapeViewer.jar [optional command line arguments]	

ShapeViewer is written in the Java language, and is therefore by default assigned a fairly small amount of system memory with which to work. Large shapes may exhaust this memory (see Displays for details on how to monitor the amount of memory used by the ShapeViewer.

You may assign more memory to the ShapeViewer when you launch the program. While each Java interpreter is may use different means to set this value, as of this writing this is typically accomplished by adding the following argument to the command which starts the ShapeViewer. Users of a batch file or a shell script to start the viewer may have edit the text of the file or script.

argument	explanation
-Xmx200M	Instructs Java to allocate 200 Megabytes (M suffix) to the java program

Command line arguments pertinent to the Java interpreter (as distinct from the Java application program) must typcially preceed the "-jar jarfile.jar" part of the command line. Details of legal Java command line arguments and syntax can generally be found in your local system documentation, and by typing

java -help

on a command line.

An example of using this argument is:

java -Xmx250m -jar ShapeViewer.jar

which runs the ShapeViewer with 250 megabytes of memory assigned to it.

A more complex example of starting the ShapeViewer from a command line is:

java -Xmx300m -jar ShapeViewer.jar -filelist showThese.txt

which runs the ShapeViewer with 300 megabytes of memory and initally displays all shape files whose names are listed, one per line, in a text file named 'showThese.txt'. In this instance, the text file "showThese.txt" is assumed to be in

the same directory (folder , for Macintosh and Windows users) from which the command is typed.

MENUS AND BUTTONS

File menu items read shapes from local computer files or from remote files.

Load Shape
: Replaces the current display with shapes read from a file on the local computer. If a file cannot be understood by ShapeViewer it is ignored. The display is centered at the midpoint of the shape.

Add Shape
: Adds a shape read from a file on the local computer to the display. The viewing angle and zoom factor of the scene are unchanged.

Load Scene
: Replaces the current display with shapes read from a scene file.

Add Scene
: Adds to the current view all shapes stored in a scene file. The viewing angle and zoom factor of the scene are unchanged.

Save Scene As...
: Saves the parameters of the current display to a XML file on the local computer. The current scene may be restored by reading the this file (See: Load Scene).

Save Scene As Image...
: Saves the current display to an image file on a local or remote computer.

Load Directory
: Replaces the current display with shapes read from files in a directory. If a file cannot be understood by ShapeViewer it is ignored.

Add Directory
: Adds to the current scene all files in a directory. If a file cannot be understood by ShapeViewer it is ignored. The viewing angle and zoom factor of the scene are unchanged.

Exit
: Exits the ShapeViewer.

View menu items change how the shapes are drawn.

Clear Viewer
: Removes all loaded shapes.

Fit to View
: Adjusts the current display zoom to show all shapes.

Perspective
: Alternately changes projection policy to 'perspective' or 'parallel'

Draw Mode

: Points
  : Draws the points of shapes.
: Lines
  : Draws the lines of shapes if possible.
: Surface
  : Draws the shapes as a wired surface if possible.
: Surface Fill
  : Draws the shapes as a solid surface if possible.

Draw Start Point
: Marks the first point of a line for shapes constructed from lines.

Draw Stop Point
: Marks the final point of a line for shapes constructed from lines.

Set Background Color
: Selects and sets the color of viewer's background.

Show Message Pane
: Alternately hides or shows the message display window.

Scene Orientation

Standard Orientations

+X
: Rotates shapes so that shapes' right sides face to the user if the user viewing from front. Similar to 'Viewing from Left'

-X
: Rotates shapes so that shapes' left sides face to the user if the user viewing from front. Similar to 'Viewing from Right'

+Y
: Rotates shapes so that shapes' bottom sides face to the user if the user viewing from front. Similar to 'Viewing from Bottom'

-Y
: Rotates shapes so that shapes' top sides face to the user if the user viewing from front. Similar to 'Viewing from Top'

+Z
: Rotates shapes so that shapes' back sides face to the user if the user viewing from front. Similar to 'Viewing from Back'

-Z
: Rotates shapes so that shapes' front sides face to the user if the user viewing from front. Similar to 'Viewing from Front'

Favorites

Select
: Loads and applys the named favorite setting.

Add...
: Adds current setting such as draw mode, view direction, light direction etc to a file, so later on user can reload it by choose 'Select' menu item.

Delete
: Deletes permantly the named favorite setting.

Lights
: Manually selects the light direction.

View Direction
: Manually selects the direction to view the scene.

Advanced menu items provide advanced user services.

Beanshell
: Show a interactive Java BeanShell terminal window. The ShapeViewer may be controlled by typing commands from its script language into this window. Java commands are also interpreted.

  As an example, type 'print(System.getProperties());' and hit Return.

  You may type the predefined commands to manipulate the viewer.

  It has a variable named 'viewer' which is the reference to the Viewer object, e.g. You can change viewer's background color to red by typing 'viewer.setBackGroundColor(Color.red);'.

Start Server...
: Starts server so other viewers can connect to this viewer as a ShapeViewer 'Client'. If you are behind a firewall you may need to set the values of the 'Listen Port' and 'Export Port' to IP ports to which your firewall allows access. If the numbers of these ports are changed from their default values you will need to to notify any client viewer users of their values.

Stop Server
: Stops a running Server.

Connect to Server
: Put your viewer into 'Client' mode and connects to a server Viewer. You must specify the server's IP address and its port number. If you are behind a firewall, the server may be unable to use the 'callback' method of updating

your client's display. In this instance you may choose to use the (slower) 'poll' method of receiving server updates.

Disconnect Server
: Disconnects a viewer from a Server.

Load Shape into a New Viewer.
: Opens a File Browser to let user select a shape, then starts the server and loads the selected shape to a new Viewer. The new Viewer connects to the server.

Help menu items describe the program and how to use it

Help Contents
: Displays a summary of the ShapeViewer menus, buttons and mouse actions

About
: Displays the revision of the ShapeViewer, it's support libraries, credits, and funding acknowledgements

View Log
: Displays a detailed log of program status, actions, and results.

Mouse actions control the viewing angle, zoom, and location of the shape display.

	Computer		
Function	**Windows**	**Mac OS X**	**Linux**
Rotate Display	Left Mouse Button	Mouse	Left Mouse Button
Zoom Display	Middle Mouse Button OR Left Mouse Button with key “Alt” down	Option+ Mouse	Middle Mouse Button
Pan Display	Right Mouse Button	Apple (cloverleaf) + Mouse	Right Mouse Button

---

Control Pane controls the rendering attributes such as color and visiblity of the shapes.

Shape Popup Menu
: Selects the shape whose color and visibility will set by the visibility controls.

Normals

- On
  : Uses the surface normal vectors to illumine a surface.
- Off
  : Ignores the surface normals.
- Flip
  : Flips the direction of the normal, if it is currently On

Transformation

- Read From...
  : Reads a linear transform from an AIR file and apply it to the currently selected Shape.
- Reset
  : Resets the transformation of the currently selected Shape to nil.

Hide Checkbox
: When checked, hides the Shape selected by the *Shape Menu*.

Color Button
: Shows a color dialog. This dialog allows a user to color a shape by either selecting a single color for the entire shape, coloring it's vertices from scalar

data attribute values passed through a user-defined color map, or from a set of RGBA colors defined by a 4-tuple vector data attribute, if any exist in the shape.

Ranges report the dimensions of the world space occupied by the shapes.

Color Map Dialog .

The color map allows a user to define a function mapping a scalar (data attribute) value to an RGBA color.

The color map dialog has 6 panels. From left to right, the first pane shows the currently set colors.
The second panel shows the values of control points of the mapping function.
The third, fourth , fifth and sixth panel define the map function in terms of it's primitive Red, Green and Blue and Opacity (alpha) color components.

The color component map functions are formed from piecewise linear maps joined at control points. Maps of any complexity may be formed by adding, deleting and moving the control points.

Add a Control Point by double - clicking the mouse in the color panel.
Delete a Control Pointby double-clicking the control point.
Change a Control Point by dragging the point, or by right-clicking the control point and editing the value dialog box.

A control point has two components, the first is its vertical value, associated with the value of the (mapped) data attribute. The second component is refers to a color component (R,G,B or A). The horizontal value ranges from 0.0 to 1.0.

The (vertical) value range of the color map can be modified by two text fields to determine the range of data attribute values colored by the color map. Data values outside the range of the color map are colored black.
This range may (and should) be manually edited to customize the coloring scheme.
Note:After editing a range value you must type type the **Enter** key to update the color map with the new value.

Custom Color Maps may be saved to and restored from disk files.

Color Maps
: Selects a color map for use.

New
: Creates a new color map.

Load...
: Loads a previously saved color map file.

Save As...
: Saves the current color map to a color map file.

Save as Image...
: Saves an image of current color map to an image file on a remote or local computer.

Rename...
: Changes name of the current color map.

Reverse
: Reverses the current color map top-for-bottom.

Button Panel

Shape
: Selects the shape to which the color map will be applied.

Attribute
: Selects the attribute of the shape to which the color map will be applied.

Ok

Applys the color map to the selected shape and selected attributes, and closes the color map dialog.

Cancel
: Closes the dialog and do nothing.

Apply
: Applys the color map to the selected shape and selected attributes.

DISPLAYS

The User interface of the ShapeViewer is composed of three display areas and a set of menus whose actions control the viewer.

The display panels show

- The current view of all selected shapes, in the same coordinate system. Clicking on a shape displays it's filename in the ShapeViewer status display (you may need to hide a surrounding shape to do so).
- The bottom panel shows status and error messages that describe the actions of the viewer. At the bottom right of this panel the Memory CCB_Tools_MeiheXu_SurfaceExtractionTutorialUsage display reports the amounts of application memory currently used, and the total available for use. An example of this

display are the values "80/259M". If the left hand number (current memory usage) approaches the value of the right hand number (total available memory) the ShapeViewer is in danger of running out of allocated memory. In this condition we recommend quitting the program and starting again with a larger amount of application memory. See Launching for more details.

- The right hand panel shows a list of the currently displayed shapes and allows changes in the display characteristics of selected shape. This panel also shows the range of the X, Y, and Z coordinates of a selected shape. Once a shape has been selected in the popup menu located near the top of panel the following display properties of that shape may be changed:
  - The shape may be hidden, or revealed (if hidden) in the view panel.
  - The use of surface normal vectors in the "Surface Fill" drawing mode may be toggled on, or off.
  - The direction of the surface normals may be reversed to improve the surface fill shading.
  - The color of a shape may be set (brings up Color and Attribute Color Map dialogs).
  - The location of a Shape's vertices may be changed by application of a AIR linear transform stored in the AIR linear transform file format.

Ranges
The Ranges report the dimensions of the world space occupied by the shapes.

COMMAND LINE ARGUMENTS

Command line arguments to ShapeViewer - abbreviated listing

Argument	Required	Default Value	Description	Notes
-help	no	n/a	Prints command line usage summary to console.	
-files file1 file2 ... file_n	no	n/a	displays listed shape file(s)	
-filelist filelist	no	n/a	displays files listed in a	expects 1 file

-filelist filelist	no	n/a	file named 'filelist'	name per line
-directory directory	no	n/a	display all files in a directory	
-scene scene_file	no	n/a	recreate a scene saved in a scene file	
-verbose	no	n/a	shows detailed information about the program's operations	
-version	no	n/a	print version number of program	
-server	no	n/a	Starts the ShapeViewer in 'server' mode	
-port	no	1099	selects a port to which clients will initially connect to request an export port.	requires '-server' option
-exportPort exportPortNum	no	n/a	selects the port used to communicate with a client	requires '-server' option
-codebase url	no	http://www.loni.ucla.edu/~librarian/shapetools/ShapeViewer.jar	URL of an RMI codebase matching this application.	requires '-server' option
-host	no	localhost	sets IP address of the controlling ShapeViewer server	requires '-client' option
-hostPort hostPortNum	no	n/a	port of server to initially contact.	requires '-client' option
-callbackPort	no	callbackPort	A port unobstructed by a firewall	Server will communicate to client via this port.
-nocallback	no	n/a	Indicates client is shielded from internet by an impenetrable firewall	A slower method of communicating with the server is used.
			Delay, in	

-updateRate seconds	no	4	seconds, between client requests to server for a status update.	Requires '-nocallback' option.
-console	no	n/a	enables writing of detailed status messages to the Console	

Complete listing of ShapeViewer command line arguments.

Argument	Type	Required	Description	Notes
-help	flag	No		
-version	flag	No		
-console	flag	No		
-sandbox	flag	No		
-nogui	flag	No		
-hiddenGui	flag	No	Don't show the GUI on startup	
-mini	flag	No	Start up with a small window	
-noAdvancedMenu	flag	No	Omit the Advanced Menu	
-points	flag	No	Start with point rendering enabled.	
-lines	flag	No	Start with line rendering enabled.	
-surface	flag	No	Start with surface rendering enabled.	
-fill	flag	No	Start with surface fill rendering enabled.	
-noExitJvmGui	flag	No	Don't exit the JVM if the applet window is closed.	
-noPreferences	flag	No	Disables use of stored user preferences.	
-plugin	flag	No	Enable Plugins	
-special	flag	No	Enable special features.	
-pluginDirectory	String	No	Read plug-ins from Jar files	Jar files names must end with ".jar" and contain Java ".class"

-pluginDirectory	String	No	in named directory.	files.
-pluginID	String	No	Plugin ID of plug-in to select on startup..	
-shapeioPluginOptions	String	No	ShapeToolsIO Plugin options	
-directory	String	No	load all files in named directory.	
-filelist	File	No	load all files listed in named file.	
-scene	File	No	load saved scene from named file.	
-out	File	No	.	
-server	flag	No		
-port	Integer	No		
-exportport	Integer	No		
-client	flag	No		
-host	String	No	Connect server to named host.	
-hostPort	Integer	No		
-callbackPort	Integer	No		
-updateRate	Integer	No		
-codebase	String	No	Set RMI codebase	Default value is: http://www.loni.ucla.edu/~shapetls/shapetools/ShapeViewer.jar
-nocallback	flag	No		
-connectClass	String	No	Set Connection class.	
-title	String	No	Set Application Title.	
-nearClipPlane	Float	No	Set location of physical eye clip plane.	
-loadSerial	flag	No	Disable concurrent laoding of multiple files.	
-loadMilliseconds	Long	No	Milliseconds allowed a shape load process. Default is 1 minute	
-debug	flag	No	Enable debugging log messages.	
-debugClasses	flag	No	Enable debugging in listed classes (comma delimited).	
-files	flag	No	A list of	

-files	flag	No	shape files	
-help	flag	No	Prints help and usage information.	
-version	flag	No	Prints version number of this application.	
-verbose	flag	No	Enables detailed status messages.	
-license	flag	No	Prints license details, and exits.	
-twiki	flag	No	Writes command line arguments as a TWiki format table to output and exit.	
-pipeline	flag	No	Write pipeline descriptor file to output and exit.	
-pipelineJar	String	No	Specify path to an application jar file, as a pipeline resource.	Example: "pipeline://localhost//home/zed/Connery.jar"
-pipelineJvm	String	No	Specify a Java Virtual Machine called by a pipeline.	
-pipelineJvmOptions	String	No	Specify Java Virtual Machine options used by a pipeline.	

SCENES

ShapeViewer can save the visible scene to a file and later , restore the saved view. Both files viewed and viewing perspectives are saved. When prompted for a file name in which to store these setting be sure to add the filename extension ".scene". Although not mandatory, files that do not end in ".scene" will not be displayed when the "Load Scene" option is selected.

DRAWING MODES

ShapeViewer supports three distinct drawing modes:

- Points: draws the vertices of a shape with no connections between them
- Lines: draws the lines that connect a shape's vertices. These connections follow the structure of the underlying shape: If the shape is a set of contour lines, for example, a set of lines will be drawn; if the shape is a connected surface then the lines will appear to delineate a (connected) surface, and so on.
- Surface: the points are connected into a regular surfce if possible. Not all

shapes can be so connected.

READING SHAPES

READING SHAPES FROM A FILE

Shapes are typcially stored in files located either on the local computer or on a remote web srver (See "Reading Shapes from a Web URL"). To display a shape stored in a disk file choose "Load Shape" from the File menu. ShapeViewer will clear the currrent display and attempt to read the file selected in the file selection dialog that is displayed. A status message is written to the message history window. In case of errors, the message is written in red.

READING SHAPES FROM A FILE LIST

Files to be read may be listed in a single text file, with one filename per line.

Multigrid files may be designated by appending ":m" to the filename.

As an example, if the following three lines are saved in a text file named "my Files.txt" then the three named (shape) files will be displayed if ShapeViewer is started by the command line

"java -jar ShapeViewer.jar -filelist myFiles.txt".

Please note that the ".txt'" suffix is optional; ShapeViewer does not require any particular file name extension to designate file name lists.

```
/home/users/tester/brain.ucff
manyGrids.ucf:m
../../myFiles/sample.ucf
```

READING SHAPES FROM A WEB URL

Shape file data may be read from a remote file by using a web URL as the filename. The remote file must be available to the remote web server that is referred to by the URL. An example URL would look like this:

```
www.loni.ucla.edu/samples/sample.ucf
```

To read a remote URL into the ShapeViewer, choose "Load Shape" from the file menu and enter the URL of a shape file. *Note:* the preceeding illustration is only an example and is not gauranteed to work)

!! VFS Browser

All files selected in the ShapeViewer use the CCB Virtual File Systems (VFS) Browser to choose the origin, or destination, of a file. Files on both the local computer and a remote computer may be accessed with the VFSB Browser. This allows the ShapeViewer to use data files stored on any computer to which you have net access. Files may also be saved to remote systems with the VFS Browser.

Files located on the local machine (on which you are running the ShapeViewer) are displayed in a familiar tree of folders and files. To access remotely stored data files you must first log in to the remote computer.

READING SHAPES FROM A REMOTE COMPUTER

follow the following steps:

1.) Select "Load Shape" from the "Files" menu
2.) Click the button titled "VFS Source", displayed at the top left corner of the VFS Browser window.
3.) from the popup window (with the mouse button held down) select first the "ftp" and then the "Connect to SFTP Server" menu item. If the remote

computer supports only the FTP protocol (discouraged) then select "Connet to FTP Server" instead.

4.) Fill out the login dialog that appears. The fields are:
    - Remote Host - the internet address of the remote computer (ex. thecla.loni.ucla.edu).
    - User - the user name with which you access the remote computer.
    - Password - the password with which you authenticate your identity to the remote computer.
    - Start Diretory - the directory whose contents you want the VFS Browser to access.
5.) After filling our the requested information, click "OK" to log into the remote computer. This may take some time, depending on your network speeds and the responsiveness of the remote system.
6.) After the display updates with the files and folders of the remote system, select the Shape file you wish to display. Please Note that only one file may be selected at a time. Wildcard characters taken literally, and their use is discouraged.
7.) The "File Name" data entry field, near the bottom of the VFS Browser form, will show the name of the file you have selected (Alternativly, you may also type the file name into this field directly).
8.) Click the "Open" button to read and display the remote data file. If this file is large and / or your intervening internet connections are slow, it may take some time to retrieve and display the remote data file.

Storing a frequently accessed remote location for easy re-use

Frequently accessed remote data locations may be stored for re-use in the future. To add a remote location to your list of 'Favorite' locations

1.) Click the 'Favorites' menu at the top of the VFS Browser
2.) Select "Add To Favorites" from the popup menu. You currently displayed remote login location will be appended to the list of favorite remote login sites.

A Favorite location may be removed from the list by selecting "Remove From Favorites" from the "Favorites' popup menu.

Logging off a remote computer

The VFS Browser maintains a list of currently accessible file systems. Once your work on a particular remote system is complete it is suggested you terminate your (remote) login session. You may do so by clicking the "VFS source" button on the VFS Browser, and then selecting the "Logout" menu choice. Click the remote session you wish to end, and you will be logged off that computer.

Logging in to a Favorite remote location

To access a location saved into the 'Favorites" remote sites menu

1.) click the 'Favorites' button at the top of the VFS Browser window
2.) Select the remote location you wish to access. The login dialog will appear, with your access information already filled out, **except for your password**, which is not stored in order to protect your privacy. Enter your password and procdeed as in an unfavored login.

MESSAGE PANE

ShapeViewer provides many status messages to inform you of its actions. These are displayed in the Message pane. This window may be hidden by unchecking the "Show Message Pane" menu item in the View menu. Messages written to this window may optionally be saved to a separate text file. To do so, choose the "Save Message History" menu from the File menu. You will be asked to name a file in which subsequent messages will be written. If you select an existing file the messages will be appended to the file.

COMMAND LINE OPTIONS

APPENDIX 1 - UCF FILES

UCF files (Universal Contour Files) are a file format used in LONI to store contours. A contour is simply a set of three diimensional points connected as a line. Contours are typically generated by tracing the intersection of an anatomical structure with an intersecting plane (MR image, for example). An anatomical structure may therefore be approximated by a collection of spatially adjacent contours. A single UCF file stores one such collection of contours. No particular restrictions of orientation, co-planarity, or number of vertices in the contours so stored are placed on the contents of UCF files. It however commonplace that the method of constructing the contours stored in a UCF file may impose shared restrictions on the file contents. As an example, the following UCF files stores 4 contour lines that are drawn in a three dimensional space.

```
<width=>
128
<height=>
128
<xrange=>
0.0 11.0
<yrange=>
0.0 3.5
<zrange=>
0.0 4.0
<levels>
4
<level number=>
0.0
<point_num=>
5
<contour_data=>
0.0 0.0 0.0 0.0
1.0 0.0 0.0 0.1
2.0 0.0 0.0 0.2
3.0 0.0 0.0 0.3
4.0 0.0 0.0 0.4
<end of level>
<level number=>
1.1
<point_num=>
5
<contour_data=>
0.0 1.0 0.0 1.0
1.0 1.0 0.0 1.1
2.0 1.0 0.0 1.2
3.0 1.0 0.0 1.3
4.0 1.0 0.0 1.4
<end of level>
<level number=>
2.2
<point_num=>
5
<contour_data=>
0.5 2.0 0.0 2.0
4.5 2.0 0.0 2.1
6.5 2.0 0.0 2.2
8.5 2.0 0.0 2.3
10.5 2.0 0.0 2.4
<end of level>
<level number=>
3.3
<point_num=>
5
<contour_data=>
1.0 3.5 4.0 -3.0
2.0 3.5 4.0 -3.1
9.0 3.5 4.0 -3.2
10.0 3.5 4.0 -3.3
11.0 3.5 4.0 -3.4
<end of level>
```

```
<end>
```

APPENDIX 2 - MULTIGRID UCF FILES

APPENDIX 2 - MULTIGRID UCF FILES

Certain projects in LONI use the UCF files to hold groups of surfaces. These special case uses of the UCF data file format are supported by the "multigrid" surface functions of ShapeViewer. The Ucf file format has no provision for determining if a file is multigrid UCF - this information must be provided by the user. When a file is designated as a multigrid file it's dimensions are checked when the file is read. Two special case dimensions are suported:

255 contour line levels of 256 points each
: The points are interpreted as a single surface grid whose points are connected by 4-sided polygons (quads).

150 contour levels of 100 points per level.
: The points are interpreted as 5 non-contgous surfaces, each of whose points are connected by quads.

APPENDIX 3 - OS X TERMINAL

The Macintosh OS X computer includes a Unix terminal program named "Terminal". In a standard installation this program is located in the folder "/Applications/Utilities/Terminal". It provides a command line interface to the Unix operating system underlying the Macintosh user interface.

APPENDIX 4 - Coloring Data with ColorMaps

How vertices are colored

If the vertices of a shape have no vertex attribute data values,then the entire shape is assigned the same color. This color may be selected, by the user, from a color palette.

If the vertices of the shape contains a single scalar value at each vertex, this number may be mapped, vi a color map, to a color which is applied to the vertex. You may pick from a predefined mapping, or create your own.

If the vertices of the shape are associated with a vector data attribute, the FIRST COMPONENT ONLY of this vector is extracted, and treated as a scalar (*version 1.1.12)*. When mutlivalent color maps are developed, these will replace this temporary means of mapping mulitvalent vertex values.

Pre-Defined Maps

Several commonly used colrom

User Defined Maps

Uses may define, save and load their own custom color maps.