SG3d

Supporting Information

1. Description

SG3d is meant to be used as a tool for viewing a data surface in three dimensions using **GRASS** on *Silicon Graphics IRIS* computers. Hardware requirements are a Z-buffer and 24 bit graphics OR a 24 bit graphic emulator, such as that on the *IRIS Indigo*.

SG3d requires a raster file to use as "elevation" and another raster file to use for surface color (or three files for Red, Green and Blue components of color). While a true elevation data file used as elevation will produce the most realistic surfaces, users are encouraged to be creative in selecting other types of data to be represented by the vertical dimension. Most continuous (as opposed to discrete) data types will result in a visualization that makes sense. While developing the program, we used chemical concentrations as the elevation file with good visual results.

Emphasis during development was on ease and speed of viewer positioning and flexibility for using a wide range of data values and sizes. Since a wire grid can be drawn very quickly, such a grid is used to provide real-time viewer positioning capabilities. Similarly, a lighting "model" provides real-time feedback as the user adjusts lighting. Grid and polygon resolution control allows the user to further refine drawing speed and detail as needed. Continuous scaling of elevation values from 1.0ee-7 to 1.0ee+ 7 provides the ability to use various data types for the vertical dimension.

In this release, we upgraded *SG3d* to allow interactive lighting specifications, vector draping, data querying (see *What's here?*), easier viewer positioning, an option to save current settings in a **GRASS** database (*3d.view*) file, animation capabilities, scale objects, labeling, an option to display lat-long data wrapped around a sphere, the ability to save images in *IRIS* rgb format files, and a few less dramatic changes such as background color options and an *animate display type* option that allows the user to view a fully rendered image while adjusting viewer positioning (see *Options...*).

2. General Notes

Environment variables *SG3D_WIDTH* and *SG3D_HEIGHT* may be set to define the width and height (in pixels) of the graphics window at startup. All *SG3d* windows may be resized using the window manager's normal controls at any time. Text in the control panels is not scalable, however, so the text will disappear if the panel is sized too small.

Before starting the program, check your current **GRASS** region resolution and dimensions. The current resolution will determine the *best* resolution available in *SG3d*. Large dimensions such as 1000 x 1000 rows and columns may cause the program to run slower. Try starting out with a smaller (i.e. 300 x 300 or less) region until you get more comfortable and experienced using the program.

After starting the program, place your **GRASS** window where it can be seen and typed into, as it is used to issue prompts and warnings and to input filenames.

Any sliders and dials used in the control panels have a FINE ADJUSTMENT: Hold down the **Ctrl** key while manipulating the dial or slider for **fine** control.

2.1. Compilation

Before compiling *SG3d*, compile the panel library and the image library by typing *make* in src.contrib/CERL/SGI/panel/9.6 and src.contrib/CERL/SGI/libimage.

When compiling under IRIX 3.x, comment out the line # *define FOUR_OH* in the externs.h file. If compiling under IRIX 4.0, leave the line in (default).

2.2. Startup

The syntax for starting the program is:

SG3d [-w] [-z] *elevfile* **color=** *colorfile(s)* **[vect=** *vectfile***] [script=** *scriptfile***] [sites=** *sitesfile***] [3dview=** *3dviewfile***]**

-w	enables writing to a script file
-z	Uses site category value as Z value (height) when displaying sites
elevfile	name of a **GRASS** raster file to use as elevation
colorfile	name of a **GRASS** raster file to use for surface color. Three raster files may be used for red, green, and blue components of color instead of a single file: **co=** *red_filename,green_filename,blue_filename*
vectfile	(optional) name of a **GRASS** vector file.
scriptfile	(optional) automatically run this script file at startup. Script file must be in user's current directory.
sitesfile	(optional) name of a **GRASS** sites list file.
3dviewfile	(optional) name of a **GRASS** 3D viewing parameters file.

If the user types *SG3d* on the command line without any arguments, the program will prompt the user for the arguments.

3. Panels

SG3d starts up with one graphics window and three panel windows visible. There are four additional panel windows that pop up when their appropriate toggle button is clicked in the *Menus* panel. To hide the pop up windows, click again on their toggle button in the *Menus* panel - using the *Quit* or *Close* options from the window manager's menu bar will cause the program to exit.

3.1. Movement Panel

Use the **Movement** panel to position the viewer over the data and to indicate viewing direction and perspective. As options in this panel are changed, a wire grid of the surface is redrawn (*fast display mode*) to immediately show the results of the changes.

XY position: The *XY position* box contains a star-shaped puck which represents the viewer. Click on the puck and drag to move the viewer around. The data may be thought of as always lying in the center of the box, with north at the top. The puck also has a dial attached to it that always points in the direction the viewer is looking.

Height: Adjusts the viewer's height.

look here: Allows the user to click on a spot on the data surface which will remain centered as the viewer position changes. Clicking on *here* results in a prompt in the **GRASS** window:

LEFT MOUSE BUTTON to mark new center of view

ESCAPE key to cancel

After clicking on the surface, the new center is automatically set and the surface is redrawn.

look center: Automatically fixes new center of view at the *center of the graphics window* (not the center of the data). If there is not a surface point at the center of the window, *look center* chooses the centroid of the visible surface for the new center. Requires no mouse input, so may be used in scripting. (see *script...*)

look cancel: Clicking on *cancel* allows the user to move over the surface with no fixed center of view (default).

Z exag: Exaggerates the vertical dimension of the surface. An exaggeration of 1.0 represents the surface with x, y, and z units all being equal. For example, if the easting and northing units are in meters and the elevation in feet, a Z exaggeration of .305 would be the *true* exaggeration for the surface. For most geographical applications, greater than true exaggeration produces more visually pleasing images.

Z Scale: May be thought of as a course adjustment for Z exaggeration. Originally included to handle data sets with vastly different units or scales used for elevation, the program now does a decent job of guessing the desired *Z scale* at startup.

Perspective: Adjusts the viewer's angle of view.

Ortho: Uses an orthographic projection instead of perspective. This type of projection tends to give an image less feeling of depth, but it can be useful when perspective causes too much distortion. If the *Perspective* slider is activated while *Ortho* is toggled on, *Ortho* is automatically cancelled.

View Dir., Twist, and View Incline: adjust the viewing direction. Adjusting the *View Dir.* or *View Incline* dials when the *look* option (fixed center of view) is on will result in the *look* option being cancelled, as if you clicked on *look cancel* first.

Reset: Sets all settings to their startup values.

3.2. Control Panel

Use **Control** panel options to select the type of surface to be drawn. Most of these options are toggle buttons that are enabled by clicking once and disabled by clicking again. After options are set, click on *Draw* to see the results. Clicking on *Cancel* while the surface or vectors are being drawn cancels the drawing. Clicking on a button other than *Cancel* while drawing will produce unexpected results.

Grid Resolution: sets number of cells between grid lines when drawing wire grids (in both *fast display mode* and *Draw* mode). A larger grid resolution number produces a more sparse grid.

Poly Resolution: sets number of cells to be used as dimensions of each polygon drawn when drawing polygons. Each drawn polygon actually represents (Poly Res x Poly Res) cells.

Polygon: draw surface using filled polygons (most realistic).

Wire/Poly: draw surface as polygons, but with wire mesh draped over it.

Wire: draw surface with a wire mesh. This differs from the mesh drawn in *fast display mode* in that hidden lines are not drawn.

Col Wire: same as Wire, but use colors from raster file for wires. Also affects grid drawn in *fast display mode* - useful when trying to position over a specific data area more identifiable by color than by elevation.

Gouraud: smooth shade surface to blend individual cell colors together.

Fringe: draw fringe around outside edges of surface.

Lights: use lighting to show surface relief.

Surface Only: (when using lights) show surface without color from raster file (renders using color from lights).

What's here? Like the **GRASS** program *d.what.rast*, you are prompted to use the mouse to mark spots on the surface that you would like more information about. After clicking on the surface, data prints to the **GRASS** window. First, the geographical coordinates are printed. Then, elevation is bilinearly interpolated from the current polygon resolution and this is printed (setting the resolution to 1 will produce the most accurate elevation). Then one line for each data file used to create the surface (two or four files, depending on number of files used for color) is printed displaying the file name, mapset, value (not interpolated, only to the nearest cell), and category. Each time you select a point, a small x is drawn on the surface at the point selected. To stop selecting points you must click the *right mouse* button.

Triangulate: Under IRIX 3.x, using the *triangulate* feature will slightly improve drawing speed. Under IRIX 4.0, there is no performance gain. When *triangulate* is selected, polygons are always broken into triangles along the same diagonal. Without *triangulate*, polygons are still drawn as triangles when all four vertices of the polygon do not lie in the same plane, but the diagonal used is determined by the surface normals at the vertices, resulting in a more natural looking surface. This option may be deleted in future versions.

No Zeros: do not draw surface where elevation data values are zero.

Draw: draw the currently defined surface.

Cancel: cancel any drawing that is currently underway.

Vect: draw the currently defined vector file. If no vector file has been specified, this button is disabled. (also see **Vect...**)

Sites: draw the currently defined sites file. If no sites file has been specified, this button is disabled. When drawn, sites appear as small (X)s draped over the surface. Using the *-z* option at startup causes sites to be drawn using the site category value as elevation instead of draping the sites on the surface. *SG3d* uses the current *Poly Resolution* to determine the size of the (X)s; for bigger (X)s, increase the *Poly Resolution* after drawing the surface but before drawing the sites. (also see **Options...**)

Quit: exit the program.

Img Dump: Prompts user in the **GRASS** window for a name to be used for an IRIS rgb format image file, then writes the contents of the graphics window to that file. This is basically a screen dump of the portion of the screen defined by the boundaries of the graphics window, so if any panel windows or windows from other applications are partially covering the graphics window, they will appear in the saved image as well. To view the image files, you can use the SGI programs *ipaste* and *imgview*. SGI also provides many programs to convert rgb image files to other formats: *toppm*, *totarga*, *tosun*, *tops*, etc.

3.3. Menus

The **Menus** panel contains buttons that toggle six additional pop-up panels. To make a hidden window visible, click once on the desired button. To hide the window, click again on the button.

3.4. Lights...

The **Lights** panel contains adjustments and options to control the position, color and brightness of the lights and the reflectivity (shininess) of the surface. There are actually two lights: one dim white light remains directly above the surface at all times and serves as fill light. The main light, which the user controls, is a "local" light which is positioned much like the view position. When lights are being adjusted, a sphere appears on the surface and is continuously redrawn to

show the effects of lighting changes. The lighting adjustments do not affect the appearance of the surface until it is redrawn. Neither light is directional; light is emitted equally in all directions.

Follow Viewpoint: By default, the light position is near the viewing position and the light moves with the viewer. Toggling this option off will keep the light fixed in one position as the viewer position changes. Adjusting the *Height* or *XY position* of the light automatically toggles *Follow Viewpoint* off.

Show Model: The sphere showing current lighting effects can be removed by toggling this option off.

Height: Adjusts light height.

XY position: Adjusts light position over surface; north is always at the top.

Light Brightness: Adjusts the brightness of the light.

R, G, B: Adjusts the Red, Green and Blue components of the main light.

Ambient Light: Adjusts the ambient light associated with the main light and the brightness of the fill light.

Surface Reflectivity: Adjusts the shininess of the surface.

3.5. Vect...

The **Vectors** panel contains adjustments for displaying vector files on the surface. After selecting options in this window, select *Vect* in the **Control** panel to draw the vector file.

New Vect: Prompts user in the **GRASS** window for the name of a new vector file. Only one vector file is "current" at any given time, but there is no limit to the number of files that may be loaded and drawn on the same surface one at a time.

Drape, Flat, Vect Z: If *Drape* is selected (default), lines are draped over the surface and *Vect Z* controls the distance above the surface that the vector lines "float". e.g., if vector lines appear to be buried under the surface in spots, try increasing the *Vect Z*. If *Flat* is selected, lines are drawn in a single plane and *Vect Z* controls the vertical position of the plane. Adjusting *Vect Z* in *flat* mode causes an outline of the plane to be drawn to help the user position the level of the plane.

Line Width: Width (in pixels) to draw vector lines.

color, Darker: Select color used to draw vector lines. Toggling *Darker* will use the selected color at a lower intensity.

3.6. Options...

Save Current 3d View: Prompts user in the **GRASS** window for the name of a file in which to save the current position and control settings.

Load 3d View: Prompts user in the **GRASS** window for the name of a *3d.view* file in which position and control settings have been saved, loads the file and resets dials and buttons. If loading a view that was saved when viewing a different region, a warning is printed describing the percentage of the region from which the view was saved that is overlapped by the current region. In such cases, loaded settings may produce unexpected results. *3d.view* files are **GRASS 4.1** database files, and the region may be changed using *g.region 3dview= viewfilename* (*SG3d* would then need to be restarted). The *3d.view* files used by this program are compatible with *3d.view* files used by the **GRASS** program *d.3d*, but minor differences in the view may occur due to the different ways the two programs interpret parameters.

New Sites File: Prompts user in **GRASS** window for a new sites file.

sites color: Select color for sites.

New Cell (color) File: Prompts user in **GRASS** window for one or three new raster files to use for surface color. The program allows the user to use a single raster file for surface color or three separate raster files for the red, green, and blue color components. The first prompt will ask *Choose single cell file OR RED file:*. If you want to use a single raster file for surface color, enter that filename and press RETURN, then press RETURN again at the second prompt *Choose GREEN file or < RETURN> for single file.*

New Elevation File: Prompts user in **GRASS** window for a new raster file to use for elevation. After loading the file, the new surface is immediately drawn using the current drawing mode, without erasing the screen first. This is not a bug. Due to memory usage, *SG3d* does not allow more than one surface to be in memory at one time, but drawing the new surface in the same visual space as the previous surface enables a careful user to view multiple surfaces simultaneously. If you only wish to view the new elevation file, you may cancel the drawing of the new surface by clicking on the *Cancel* button, just below the *Draw* button. Any subsequent drawing will only show the new surface. If the range of data in the new elevation file is different than the range of the previous elevation file, it may be necessary to select *Reset* from the **Movement** panel for the surface to be rescaled for easier viewing. When displaying multiple surfaces, if you want to use a different raster file for color you must load the new color file first, then the new elevation file. There is no limit to the number of surfaces that can be displayed in succession, but once *Draw* is selected and the surfaces are erased, only the latest surface is available.

animate display type: Toggling this button will cause the current display type (*Polygon*, *Wire/Poly*, or *Wire*) to be drawn instead of the *fast display mode* wire grid when the user is adjusting **Movement** panel controls. These drawing modes take considerably more time and the display is double buffered, so drawing progress is not shown. When *animate display type* is toggled on, the program checks currently specified resolution and issues a warning if drawing time will be very slow. This button was hidden on the **Options** pop-up panel to prevent its accidental use because it can be time consuming and frustrating if you have the resolution set too fine and try to animate a fully rendered surface of say a quarter of a million polygons (500 x 500). However, including this option allows those with very fast graphics systems to take advantage of their hardware. One further note: double buffering causes available color bitplanes to be halved, so the image quality while animating will be degraded. When you release your mouse button after adjusting viewing position, the image is redrawn in single buffer mode utilizing all color bitplanes. This effect is especially noticeable on systems with only eight bitplanes.

background color: select color for background. The program chooses a contrasting color to use for the grid in *fast display mode.*

map lat-long to globe: This button only appears when the current **GRASS** location uses a latitude-longitude projection rather than UTM or x-y. Toggling this button ON causes the lat-long data to be mapped to a sphere. Drawing in this mode is noticeably slower because each point has to be transformed to spherical coordinates. This is a relatively new feature and still has some bugs. The following functions of SG3d are not available when the surface is mapped to a sphere: *look here, What's here?, Fringe,* and none of the functions on the **Scale** panel. In addition, viewer positioning functions are not as intuitive in this mode. By default, the viewer looks toward the center of the sphere, but you may use *look cancel* followed by *View Incline* and *View Dir.* to let you look in a different direction. The *look center* button always resets the center of view to the center of the sphere when using this mode.

Scale... & Label... These two buttons toggle additional pop up panels.

3.7. Scale...

The **Scale** panel contains adjustments for displaying a scale object anywhere on the surface or ruler marks along the edges of the surface. Scale objects available are wire outlines or solid representations of a square plane or a cube showing z-exaggeration effects and an arrow pointing north. This panel is disabled when using *map lat-long to globe* mode.

place scale object: Prompts user in the **GRASS** window to use the left mouse button to mark center of scale. Click with the left button in the graphics window at the point where you want to place the object. If the point is on the surface, the object will be placed so it is resting on top of the surface. If the point is off the surface, the object will be placed where the position clicked intersects with the plane of constant z value at the minimum z value of the surface (or minimum non-zero, if *No Zeros* is toggled ON). This method of determining the elevation of the scale object may be overridden (see *elev* below).

solid, wire, north arrow, cube, plane: Combinations of these toggle buttons determine the appearance of the scale object.

draw ruler: Draws ruler marks along the selected sides of the data at intervals determined by the value of *size*. For a graded ruler: draw ruler once, reduce *size* by an even multiple, draw ruler again, reduce *size*, etc. Note: The width of the ruler lines may be changed by adjusting *Line Width* on the **Vectors** panel.

NW, SW, NE, SE: Determines which sides of the data the ruler will be drawn on and the direction from which the scale marks begin. For example, if *SW* is toggled on, ruler marks will appear on the south and west sides of the surface, proceeding from the southwest corner. If both *SW* and *NE* are toggled on, ruler marks will appear on all four sides of the surface, proceeding from the southwest corner and northeast corner.

elev, auto: When *auto* is toggled ON, the type-in box displays the elevation, in units of the raster file used for elevation, at which the scale object was placed. You may type in a different elevation by clicking in the type-in box, entering the elevation at which you want the object displayed, then pressing return. This automatically toggles *auto* off. Subsequent placement of scale objects will cause the object to be drawn resting on a plane of constant z value at the new elevation (also see *place scale object*).

size: This type-in box determines the size of scale objects and the distance between tick marks on the rulers, in units of the data (meters for UTM, degrees for lat-long). By default, *SG3d* chooses some power of ten times 25, 50, or 100 which results in a desirable scale size.

color: Select color for scale object and ruler lines.

3.8. Label...

The **Label** panel contains adjustments for displaying a text string in the graphics window.

place label, undo: The *place label* button prompts user in **GRASS** window to use the left mouse button to mark first letter of label. Click with the left button in the graphics window at the point where you want to place the label. The bottom left corner of the first letter of the label will begin where you click. If you are not satisfied with the positioning or size, press the *undo* button and the label will be removed.

text: Enter the text you want to place in the graphics window in this type-in box.

size: Enter the point size of the text you want to display. *SG3d* chooses the closest point size equal to or less than the size requested that is available in the font requested. One inch is equal to 72 pts.

color: Select color for label.

Times-Roman, Helvetica, Italic, Bold: Combinations of these toggle buttons determine which of eight possible font styles is used.

3.9. VectPath...

The **Path** panel contains controls for using a **GRASS** vector file to determine a path for viewer movement over the surface. By default, *SG3d* limits the number of points used from a vector file to the first 1000 points. This number is defined in path.c as MAX_PATHSTEPS, so may be changed if needed and the program recompiled. The user may request any number (within reason) of total frames once the vector file points are loaded; path points between vector points are interpolated using a cardinal cubic spline or simple linear interpolation, as requested by the user. Closed, looping paths are supported by both methods of interpolation. A path is considered to be a loop if the beginning and ending points in the vector file are identical. (Also see *Keyframe...* and **Hints - Animation**.)

new vect path: Prompts user in **GRASS** window for the name of a vector file from which to generate a path. When the path is created, the number of points used from the vector file is displayed in the *total frames* box (see *total frames*).

Follow Terrain, height, smoothing: With *Follow Terrain* toggled on, the path uses the value in the *height* type-in box to determine the z value of the viewing position (z value = *height* + elevation of terrain under path position). If *smoothing* is toggled on, the *height* value is used to determine the z value of the viewing position ONLY at the points along the path that were obtained from the vector file; any interpolated path points have their z values interpolated as well.

Constant Elevation, elevation: With *Constant Elevation* toggled on, the path uses the value in the *elevation* type-in box as the z value of the viewing position. By default, the maximum value in the raster file used for elevation is placed in the type-in box.

Look Forward, Tilt: With *Look Forward* toggled on, the viewing direction along the path is automatically set to look toward a forward point along the path. The value in the type-in box below *Look Forward* is used to determine how many points ahead to look. A negative value in the type-in box may be used to look backward. Use the *Tilt* dial to look more toward the ground or the sky when looking forward. Moving the dial in a counterclockwise direction will set the looking position more toward the ground. This effect is not displayed unless the path is running. If *Look Forward* is toggled off, the direction of view is determined by the settings in the **Movement** panel.

step, run, Frames: The *Frames* slider may be thought of as a time line representing the path. When a path has been loaded, clicking on *run* will move the viewer down the path, redrawing the wire frame surface at each path step. The user may cancel a running animation at any time by clicking on the *Cancel* button on the **Control** panel. The *Frames* slider shows the progression of the animation. Use the *step* arrows to step forward or backward one frame at a time, or move the *Frames* slider to the desired point in the animation. The number to the right of the *step* arrows is the frame number being displayed.

total frames: The value in this type-in box is the total number of frames in the animation. When a vector file is first loaded, the number of points in the file (up to 1000) will be the number of frames. Typically, to produce smooth animations, the user should use a vector file with 10 - 50 points, then change total frames to 100 or more, depending on the length of the animation desired. Requesting an excessive number of frames (i.e., 500,000) may cause memory problems.

show path: Toggling this button ON will cause the path to be drawn each time a new frame is drawn. The color and thickness of the path line may be changed using adjustments on the **Vectors** pop up panel. Showing the path can be useful while previewing and adjusting the path.

show vect: Toggling this button ON will cause vectors to be drawn each time a new frame is drawn, if a vector file has been loaded. (see also *Vect...*).

spline, linear, tension: These controls affect the interpolation method used when the value of *total frames* is different than the number of points in the original vector file

used as the path. The points from the vector file are always used as control points from which the positioning information for the total number of requested frames is derived. If *spline* interpolation is selected, the user may adjust spline tension (how much slack there is in the path) using the *tension* slider. Toggling *show path* ON prior to adjusting tension allows the user to judge how much tension to use for a smoother path.

send path to keyframes: This button reads the currently defined vector path and copies it to the **Keyframe** panel. Keyframes already defined in the **Keyframe** panel are overwritten. (See **Keyframe...**.)

run and save images: Prompts user in the **GRASS** window for a prefix to be used to name a series of rgb format image files, one for each frame. The animation is then run from the beginning, with each frame being fully rendered using the current display type and resolution (see **Control** panel). If *show vect* is toggled ON, the vector file is also drawn, then the finished frame is written to an rgb file. If files already exist with the same name, they are overwritten. You may cancel *run and save images* at any time by selecting *Cancel* from the **Control** panel (it may take several clicks on *Cancel*). This entire process may take several hours of computer time for a 100 frame animation of a complex surface, but requires no further intervention from the user. The finished rendered animation files can be played back using SGI's *movie* program if it is a very short animation or you can capture the images on video. (Also see *Img Dump* on the **Control** panel and **Hints - Animation**.)

3.10. Keyframe...

The **Keyframe** panel contains controls for producing custom animations by moving the viewer around the data surface. With keyframe animation, the user precisely defines a few positions along a movement path (keyframes), specifies the total number of frames desired in the animation, and then lets the computer generate the viewer position along the path for all the in-between frames, according to a selected interpolation method. (Also see *VectPath...* and **Hints - Animation**.)

step, run, Key Frames: The *Key Frames* slider may be thought of as a time line representing the animation. When three or more keyframes have been defined (two or more if using linear interpolation), clicking on *run* will move the viewer along the path beginning from the current position of the *Key Frame* slider bar, redrawing the wire frame surface at each path step, and advancing the slider bar. The user may cancel a running animation at any time by clicking on the *Cancel* button on the **Control** panel. The number to the right of the *step* arrows is the frame number being displayed. The *Key Frames* slider has small black tick marks representing each frame of the animation and larger blue arrows above the slider that represent keyframes (none of the latter exist until keyframes are created - see *add a keyframe*). The keyframe arrows can be grabbed with the mouse and moved along the slider. To delete a keyframe, grab the arrow representing the keyframe to be deleted and slide it completely off the slider to the right-hand side, then release the mouse button. Use the *step* arrows to step forward or backward one frame at a time, or move the *Key Frames* slider bar to the desired point in the animation. If the user clicks in the slider at a point to the left of the first keyframe or to the right of the last keyframe, or if there have not been enough keyframes defined to specify a path, current viewer positioning settings are not changed.

total frames: This is a type-in box containing the number of total frames requested. To change, click in the box, edit the value using the keyboard, and press return. The path will be updated and the *Key Frames* slider will be redrawn showing the number of frames requested as black tick marks along the slider.

add a keyframe: Adds a keyframe at the point in the animation where the keyframe slider bar is positioned. If there is already a keyframe at that point, the user will be asked in the **GRASS** window whether the old keyframe should be replaced.

clear all keyframes: Deletes all keyframes currently defined.

show path: When toggled ON, this button causes the currently defined path to be drawn as a line with (X)s marking keyframes. It also changes the display to show orthographic views of the path from the top, south and east in three separate viewports along the bottom of the graphics window. The viewer's current position is shown in these small viewports by a red dot. The color and width of the line used to show the path can be changed using adjustments in the **Vect...** panel.

show vect: Toggling this button ON will cause vectors to be drawn each time a new frame is drawn, if a vector file has been loaded. (see also *Vect...*).

linear, spline, tension: These controls affect the interpolation method used. The keyframes are always used as control points from which the positioning information for the total number of requested frames is derived. If *spline* interpolation is ON, the user may adjust spline tension (how much slack there is in the path) using the *tension* slider. Toggling *show path* ON prior to adjusting tension allows the user to judge how much tension to use for a smoother path.

run and save images: See *run and save images* on the **VectPath...** panel. This button does the same thing, but for the keyframe-defined path.

3.11. Script...

The **scripting** panel contains controls for reading and (if enabled) writing script files that contain commands to the panel system. To enable writing use the **-w** option at startup or set the environment variable *WRITE*. Script files are not **GRASS** database files, and are read from and written to the current directory.

script file: This is a type-in box that contains a default filename, *panel.script*. To replace with your own filename, click in the box, edit the name, and press return.

begin read, stop read: Use to read the script file indicated. The program will perform the actions saved in the script with nearly the same pacing as when they were written to the file, as if somebody was using the mouse (but you don't see the mouse cursor move). The *READING* button will be lit while the actions in the file are being displayed. You may still adjust settings with the mouse while a script is being read, but some actions will produce unexpected results. (e.g., if you pop up a panel then the script tries to pop up the same panel, it actually hides it instead)

begin write, begin append, stop write/append: Use *begin write* to begin a new script file. After clicking on *begin write*, any mouse actions in the panels will be recorded in the script file until *stop write/append* is selected. *Begin append* will cause mouse actions in the panels to be appended to an existing specified script file until *stop write/append* is selected. The WRITING button will be lit while the actions that occur in the panels are being recorded. When writing script files for demos, avoid actions that require user input (*new cell file*, *new vect*, *what's here*, *look here*, etc.), as only those actions that occur in the *PANELS*, not the graphics window or **GRASS** window, are recorded in the script. Also it is a good idea to begin each script writing session by pressing *Reset*, so that what you see as you write the script will be the same as when it is read (no initial conditions are assumed or reset when reading a script).

4. Hints

4.1. Lighting

Lighting greatly enhances the level of detail that can be seen in the surface. For good lighting effects, position the light so that it creates a tilted half-moon on the sphere. Usually placing the light in the N-W corner when viewing from the south works well because of the way humans tend to interpret shading. To reveal very subtle gradual elevation changes, experiment with

light placement and surface reflectivity. Keep resolution higher (3 or 4) while adjusting viewing position and lighting so that drawing time is fast. Then lower polygon resolution to 1 or 2 for the final rendering of the surface.

When rendering frames for animation, it is usually better to position lights so that they do not move with the viewer position (toggle *Follow Viewpoint* OFF).

4.2. Animation

Several methods of creating rendered frames for animation are available using *SG3d*; you may define a vector path, use keyframes, or use a panel script. What follows are a couple of examples to help you get started with animation.

4.2.1. Keyframes

Using keyframes is one of the easiest and most satisfying methods, it just takes some getting use to the *Key Frame* slider. For practice, just try making a circular path while always looking at the center of the data:

1) use *look here* to mark a fixed center of view in the middle of the surface.

2) move the puck in the *XY position* box to the southeast corner, adjust *Height* and *Perspective* until the data surface fills your graphics window.

3) pop up the **Keyframe** panel.

4) Click on *add a keyframe*.

5) position the *Key Frame* slider bar to the next place to add a keyframe by clicking a half inch or so farther along the slider.

6) move the *XY position* puck to the northeast corner.

7) Click on *add a keyframe* and repeat step 5.

8) move the *XY position* puck to the northwest corner.

9) Click on *add a keyframe*.

You should now have enough keyframes marked to define a path, so try clicking on *show path* and then *run* (NOT *run and save images*). If nothing happens, clear all keyframes and go back to step 1. Now let's complete the circular path:

10) repeat step 5.

11) move the *XY position* puck to the southwest corner.

12) Click on *add a keyframe*.

13) Now to close the loop, move the *Key Frames* slider down to frame 1, so that the first frame is displayed, release the mouse button, then click once on the slider in the empty part, past the last keyframe, and click on *add a keyframe*. This essentially copies the first keyframe to the fifth & last keyframe.

Now click on *run*. When the animation stops, adjust the spline tension or use linear interpolation, change the number of total frames, and run again. If you wanted to render the animated frames to .rgb files, you would now toggle *show path* OFF and click on *run and save images*, but why not wait until you get a little more creative - image files use a lot of disk space.

Changing keyframes: With the above keyframes loaded, try changing a keyframe as follows:

1) Position the *Key Frames* slider so that it aligns with the arrow marking the second keyframe.

2) Using the *Height* adjustment in the **Movement** panel, raise your viewing position. Notice that if you have *show path* toggled ON, you can see how your viewing position changes relative to the defined path by watching the south or east path views. The small red dot represents your current position.

3) Click on *add a keyframe*. You will be asked in the **GRASS** window if you want to replace the old keyframe: type 'y' and 'return'.

4) Run the animation again to see how the path has changed.

5) Delete the middle keyframe: grab its arrow marker (using your left mouse button), drag the marker completely off the slider to the right, and release the mouse button. Run again to see how path has changed.

6) Set total frames to about 50, then one at a time, grab the center keyframe markers and slide them toward the right so that they are spaced something like:

1.....2..3..4 Run again, notice how movement speed was affected.

7) Reverse the direction of travel by swapping the positions of the two interior keyframe markers.

Parameters stored in the keyframe path include viewing position, looking direction, twist, and perspective. The looking direction will be overridden though, when the path is running, if *look here* is active. To see this effect with the above path loaded, select *look here* and click on a corner of the data, setting a new fixed center of view. Now run the path again. Click on *look cancel* and run again. Adjusting the *Twist* parameter when creating a path will give your animation more of a fly-by effect by simulating banking turns.

4.2.2. Panel Scripts

While vector paths and keyframe animation are good for showing movement around your data surface, often animation can also be used to show change over time, or perhaps change in surface properties due to a changing environmental parameter. In such cases, you will be creating multiple grass data files to represent the change, using your own program or other **GRASS** data analysis tools. *SG3d* can still be used to produce a 3d surface animation.

Suppose you have generated 30 different raster files that show diminishing rainforest cover over a period of years. For this example, we will assume you will always be draping them over the same elevation file. After setting a good viewing position and adjusting lights, you *could* make an animation by this process: select *New Cell (color) File* from the **Options** panel, load the file, *Draw* the image, select *Img Dump*, name the image file, and repeat (30 times).

An easier way to accomplish the same thing would be to produce a panel script to do all the work for you. But if you review the discussion of the **Scripting** panel, you'll notice that we said to avoid selecting options that require user input (e.g. *New Cell File*) since input in the **GRASS** window is not recorded in panel scripts. The way around this is to know what questions the panel script will be asking and provide all the answers in a text file that will be redirected as input to *SG3d* at startup. This is kind of messy, but it works great if you're careful. So for the above example, the text file would look like this:

rast01.file< return>

< return>

img01.rgb< return>

rast02.file< return>

< return>

img02.rgb< return> ...

rast30.file< return>

< return>

img30.rgb< return>

For this example, you would create the panel script as follows: Start *SG3d*, using the -w option to allow script writing. Pop open the **scripting** panel and the **Options** panel. Enter a name for your script in the type-in box on the **scripting** panel. Now click on *begin write, New Cell (color) File, [< enter> in GRASS window], [< enter> again in GRASS window], Draw* (wait for surface

to draw), *Img Dump, [< enter> in GRASS window], begin read, stop write.* This creates a panel script that is an infinite loop, but the program will terminate when EOF is received in standard input from your text file.

Finally, you would create the animation itself by starting the program with: *SG3d elev.file co= rast01.file < your.txt.* Adjust lighting and positioning (if you have a 3dview file you want to use instead, load it at startup with *3dview= file* because if you try to load it once a text file has been redirected to the program, *SG3d* will use the first line in the text file as input in the **GRASS** window). Then pop-up the **scripting** and **Options** panels, enter the name of your panel script, click on *begin read* and take a lunch break.

You may want to save panel scripts created as above, since they could be used with different text input files to create completely different animations. On the other hand, once you've got the hang of it the panel scripts are quite easy to make. Using this method, it is also possible to "morph" a surface by loading changing elevation files instead of color files. Once you get used to this method, you can try adding multiple vector or site files to your surface, animating multiple surfaces, or combining movement with a changing surface.

4.2.3. Vector Paths

The discussion of vector paths in section **3.9** is fairly detailed, but here are a few more suggestions.

Use the **GRASS** program *v.digit* to digitize a path you would like to follow, using a minimum number of points. Or, use *v.digit* to create a subset of a roads or streams vector file, ending up with a single road or stream in a new vector file to use for a path.

Using *Look Forward* with the number of frames to look ahead set to 5 or 6 for a 50 frame animation gives good results; looking ahead only a frame or two will usually cause a lot of jerkiness.

Adjust the *Tilt* dial while previewing the running animation.

When adjusting spline tension with *show path* toggled ON, It may be necessary to temporarily adjust your viewing position to a place well above your surface, looking down, in order to get a good view of your path.

4.2.4. Image File Management

Because .rgb image files can end up taking a lot of disk space, users should be mindful of disk storage capacity when attempting to create an animation. Keeping animation files and image dumps in directories dedicated for such files helps in management. A single rgb image usually takes between 200K to 3M bytes of storage, depending upon size and complexity. To save disk space, use the SG3D_WIDTH and SG3D_HEIGHT environment variables to limit the graphics window to known dimensions. If you use the SGI *movie* program to replay animations, each frame of the animation must have been saved to files of identical dimensions; so especially if you are creating an animation piecewise, at different sessions, it becomes important to use the environment variables to maintain a consistent size for the graphics window. With the *movie* program and with other screen dump animating programs, there is a zoom option to enlarge the animation images at time of playback. So if you want the animation to fill your screen, you could set the width and height of the graphics window to half or a third their playback sizes when writing the image files, then use the zoom option (this will cause a loss of resolution though, since the zoom just replicates pixels). If you are making an animation to be recorded onto video tape, it may be necessary to set the width and height to specific dimensions.

New SGI programs released with IRIX 4.0.5, *moviemaker* and *movieplayer*, are also very useful for animations. *Moviemaker* allows you to create movie files from any number of rgb files at various frame rates. With *movieplayer*, much longer animations may be replayed than with *movie* since the frames are read directly from disk at run time rather than having to be loaded into memory.

5. Credits

SG3d written by Bill Brown and David Gerdes, U.S. Army Corps of Engineers Research Laboratories, Champaign, Illinois.

This report written by Bill Brown, Fall 1992. Your comments and suggestions are appreciated.

****

Bill Brown

USACERL - ECA

PO Box 9005

Champaign, IL 61826-9005

e-mail: brown@zorro.cecer.army.mil

phone: (217) 352-6511 ext 676

****

--

--