Stanford Exploration Project, Report 67, January 13, 1998, pages 1–164

Stanford Exploration Project, Report 67, January 13, 1998, pages 1–164

X3D (extensible 3-D visualization) update

Rick Ottolini[1]

keywords: *graphics, three-dimensional*

ABSTRACT

The X3D graphics system has been upgraded to XView user interface with the additional graphics functionality of illuminated surfaces and transformable raster images.

INTRODUCTION

In SEP–61 I described the X3D graphics system—XWindows, extensible, three-dimensional graphics system for geophysical data visualization. Its purpose was to bring *usable* 3-D graphics to SEP in two forms: canned programs that can read and plot standard SEP datasets and a set of subroutine libraries for easily writing new programs. X3D was motivated by that existing visualization packages require special hardware generally inaccessible to SEP and are not easily integrated into SEP's processing system.

The properties of X3D include:

- Three-dimensional vector and *raster* (new) graphics models.
- Extensibility. One can create new applications by adding custom parts to the generic system.
- Portability. X3D is written in XWindows, XView, vplot, and SEPlib and runs on all our computers.

There are two kinds of improvements to the version of X3D described in SEP–61. First, the user interface was upgraded from menus to control panels. This was made possible by a standard control interface called *XView* in the current release of XWindows (11.4). Second, new graphics abilities such as lighted surfaces and transformable raster arrays were added.

[1]**email:** not available

NEW CODE STRUCTURE

Layered approach

There are three layers to X3D:

- Canned applications that display datasets in certain ways. These include the applications *Cloud*, *Cheops*, *Cube*, *Flow* and *Surf* described in SEP–61.
- Subroutine libraries for custom applications. The canned applications use these routines.
- Output drivers for various interactive and hardcopy graphics devices.

User interface

Launching an X3D application puts two or three windows on the screen:

- The main canvas (Figure 1). This conforms to XView/OpenLook standards with a menubar above the canvas and message area below. The menubar contains system and graphics functions common to all applications including: *quit, vplot, refresh, double buffer, fill, draw bounding box, draw axes* and *status.*
- The 3-D control panel (Figure 2). 3-D functions include rotation, size, shift, lighting angle, and animation.
- An optional custom control window (not shown). This contains special controllers for a custom application.

An application recognizes the command line arguments:

```
size=0.7         fraction of window size
orient=25,25,0   initial x,y,z orientation in degrees
box=no           draw bounding box
axis=yes         draw axes
fill=yes         fill surface patches
buffer=yes       double buffer
resolution=2     raster resolution
out=default      hardcopy file
```

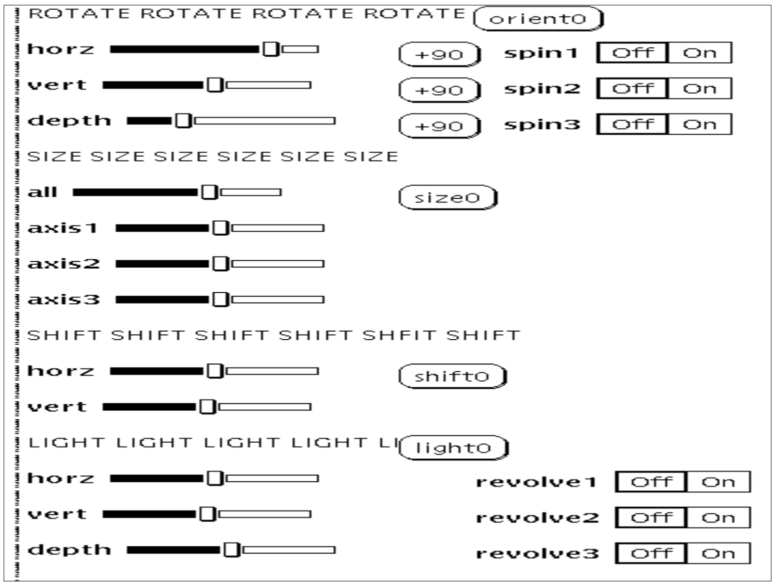

Figure 1: Three-dimensional controls. $\boxed{\text{rick1-fig1}}$ [NR]

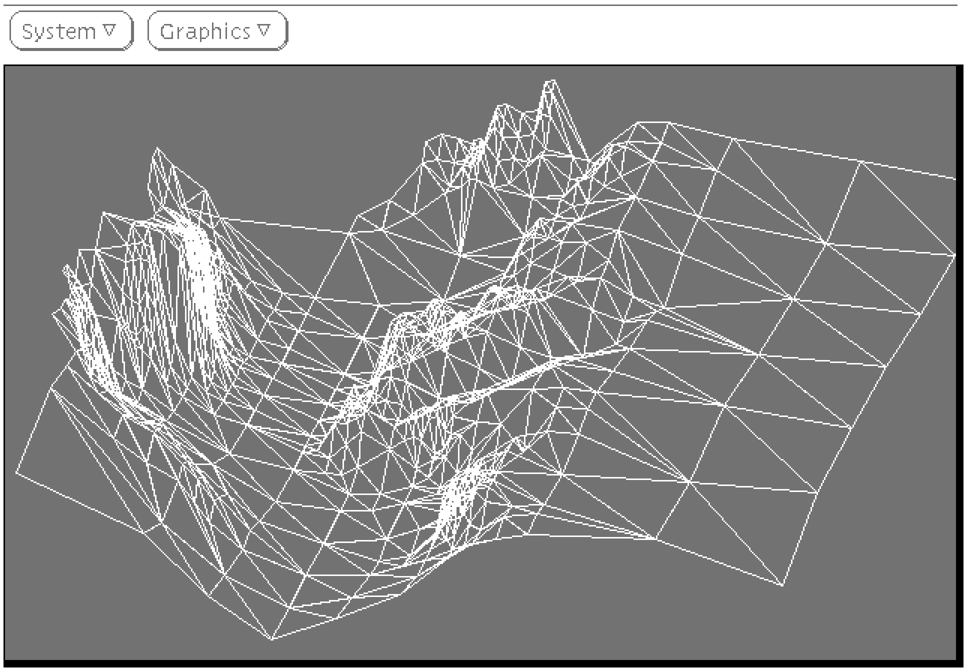

Figure 2: X3D canvas. $\boxed{\text{rick1-canvas}}$ [NR]

Custom application

To write a custom application the programmer must supply the following three routines:

- *myinit()*: called once at the start of the application. Recommending for fetching parameters, reading data from files, initializing display variables, and adding custom controls.
- *mydraw()*: called at the start of the program and each time a display control is changed. This routine redraws the entire display object in user coordinates. The drawing is initially scaled to fit the window.
- *mycommand*(command,value): Receives all command codes and optional values.

(Customization is the same as SEP–61 with one major change: The file *mybutton* for configuring control interfaces has been replaced by calling *UICommandAdd()* within *myinit()*.)

These are the graphics and user interface routines available to the custom routines:

```
*GraphicsSetRotate (x,y,z)
*GraphicsSetScale (x,y,z)
*GraphicsSetShift (x,y,z)
*GraphicsSetLight (x,y,z)
*UICommandAdd (com_type,control,label,code)
Erase ()
PointList (vert,nvert)
LineList (vert,nline)
PolyLine (vert,nvert)
Polygon (vert,nvert)
*PatchList (vert,npatch,patch_size)
*Mesh (vert,nvert,patch,npatch,patch_size)
*Raster2D (where,width,height,array)
*RasterList (where,width,height,array,npatch)
Text (where,text)
TextAlign (halign,valign)
Fat (fat)
Color (color)
*Shade (shade)

com_type = LABEL, TEXT, MENU, ITEM, SLIDER;
control = MENUBAR, CONTROL3D, CUSTOM;
Vector3D vert[], where[];
string label, code, text, color;
```

```
float x, y, z, width, height, shade;
int nvert, patch[], patch_size, npatch, fat, width[], height[];
byte array[][];

* added or changed since SEP--61
```

DISCUSSION

I expect some of the inner core to replaced by XWindows-PEX when it becomes generally available in 1991. This should not affect the custom programming interface much, if at all.

REFERENCE

Ottolini, R., 1989, X3D: extensible, interactive three-dimensional geophysical data visualization: SEP–**61.**, 319-326

.