Real Time Visualization

Gfxsanity Information

This document provides information about gfxsanity.cpp, a complete volume rendering program written by Real Time Visualization. Gfxsanity integrates a Windows NT window with OpenGL and VLI. The program code for gfxsanity is included on the distribution CD-ROM and can be adapted for use in your own applications.

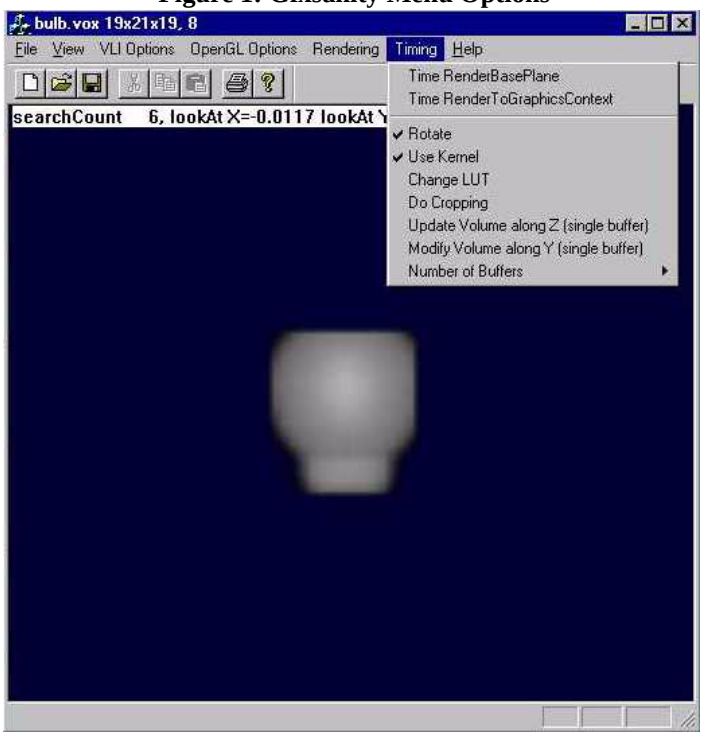
Gfxsanity provides a window with menu options that exercises certain VLI functions, such as VLI options, OpenGL options, buffering, and modifying lookup tables and crop boxes. Using these menu options with sample datasets can give you an idea of how these might work if you used similar code in your own application.

Requirements

To run gfxsanity, your system must have the VolumePro board and software installed.

Menu Options

Figure 1: Gfxsanity Menu Options



File

Open	opens a file
Internal volume	creates a new vox file. Opens a dialogue box that can set the following: • size in x, y, & z (cannot be greater than 256) • start & end voxels • data managed by either VLI or application • volume data type (8-bit, 12-bit low, 12-bit high)

View

Toolbar	Show or hide toolbar
Statusbar	Show or hide status bar
Zoom In	Zoom closer
Zoom Out	Zoom farther away

VLIOptions

Adjust Opacity	Adjust opacity table based on viewing angle
Set Supersampling	Opens a dialog box that allows you to • set supersampling factors in x, y, and z direction • set supersampling in camera or object space • set accumulation mode to grow or blend base plane
Show multiple buffers	Display multiple (intermediate) buffers in blue
RGBA	Red first, alpha last
BGRA	Blue first, alpha last
ARGB	Alpha first, blue last
ABGR	Alpha first, red last
Front to back	Blending mode set to front-to-back
Max intensity projection	Display voxels with maximum intensity
Min intensity projection	Display voxels with minimum intensity
Keep volume locked	Keep volume locked on VolumePro board
Lock volume when animating	Lock volume before each rendering; unlock afterwards. DO NOT USE FOR TIMING
Keep volume unlocked	Never lock volume
Lookup table size	Choose table size of: 4096, 2048, 1024, 512, or 256

OpenGL Options

Use OpenGL	Use OpenGL to render and transfer base plane
Update Texture	Update texture memory from base plane

Display Base Plane	Display un-warped base plane in green. The display appears in the upper left corner of the window.
Display Volume	Display volume rather than base plane.
Display Outline (VLI)	Display VLI-computed outline in addition to base plane or volume.
Display Buffering (OpenGL)	Use OpenGL Double Buffering if checked, else use single buffering
Display Volume Corners	Display corners of volume.

Rendering

Animate	Animates volume; provides a jittery rotation.
Step back one	Move back one step in animation sequence
Step forward one	Move forward one step in animation sequence
Set Animate parameters	Opens a dialog box that allows you to set certain parameters for a customized animation. These parameters include: • Axis of rotation in x, y, and z directions • Current angle • Step size (degrees) From this window, you can do animate the volume with the following options: • Step forward • Step backward • Render at this angle • Animate forward • Animate backward • Stop animation
Search 1	Causes a smooth, very fine rotation
Search 2	Search pattern different from Search 1; also causes a smooth, very fine rotation
Single step 1	Both options more forward one step
Single step 2	
Mapper test	Tests the **`VLIVolume`** class function **`MapVolume`**
RenderBasePlane	Use the **`VLIContext`** class functions **`RenderBasePlane`** and **`FetchBasePlane`** to render volume
RendertoGC	Uses the **`VLIContext`** class function **`RendertoGraphicsContext`** to render volume
Reset Search Counter	Resets search counter

Timing

Time RenderBasePlane	User VLIContext class function RenderBasePlane in a double buffered timing loop
Time RendertoGraphics-Context	Use VLIContext class function RendertoGraphics-Context in a multibuffered timing loop
Rotate	Rotate volume
User Kernel	If selected, use kernel pixel buffer directly. Otherwise, copy to user buffer.
Change LUT	Change color lookup table every frame
Do Cropping	Change crop planes every frame
Update Volume along Z (single buffer)	Update the Z slice of volume per frame FOR SINGLE BUFFERING ONLY
Modify Volume along Y (single buffer)	Map and modify one Y slice of the Volume per frame FOR SINGLE BUFFERING ONLY
Number of buffers	Choose to use 1, 2, 3, 4, 5, 6, 7, 8, or 16 buffers