AVS 5.5 FOR SUN SUNOS5 (SOLARIS)

CHAPTER EIGHT

Introduction

This chapter describes AVS 5.5 on Sun Microsystem's SPARC-series workstations running SunOS 5.7 (also called Solaris 2.7 or Solaris 7). It covers hardware and software prerequisites needed to run AVS; differences and special considerations for running AVS on this platform; and known problems.

AVS 5.5 NOTE: The supported operating system has been updated from SunOS 5.5 to SunOS 5.7 (also called Solaris 7) and related changes have been made in the versions of the supporting graphics libraries. OpenGL stereo and the Elite graphics adapter are now supported.

NOTE: Additional information on using AVS under X windows, general programming considerations, and bugs fixed in AVS 5.5 can be found in the *AVS 5.5 on UNIX Platforms* chapter above.

Hardware Prerequisites

Configurations

Sun workstations usually exist in a network environment that can be configured in a variety of ways.

- A system can be *diskless*, meaning that it executes programs locally, but gets all of its software and swap space from a server machine.
- A system can be *dataless* where it executes code, swaps locally, and has its own local kernel, but gets its */usr* partition from a server.
- A system can be *standalone* where it executes and swaps locally, and has its own copy of the kernel and the */usr* software.

However, a remote server may be providing bulk storage, such as the space required to store AVS.

- A system can be truly standalone, where it performs all functions and has all storage local.

- A system can be a *server* that may not execute AVS at all, but provides AVS to some combination of the above systems. These systems may have different, heterogeneous kernel architectures.

Some aspects of AVS installation, licensing, and use require that you be aware of the network configuration under which you are trying to use AVS. These "Installation Notes" make frequent use of phrases such as "on which AVS will execute," "on which AVS swaps," and "on which AVS is stored." Note these phrases when they occur as they are critical to correctly configuring your system to run AVS.

SPARC Models

AVS will execute on any Sun SPARC-series workstation supported by SunOS 5.7 and later. At present, this means machines with *sun4c*, *sun4m*, *sun4e* and *sun4u* kernel architectures. Typing the */usr/bin/showrev* command will produce a listing that describes the kernel and application architecture of your hardware such as the following:

```
Hostname: seymour
Hostid: 807853f9
Release: 5.7
Kernel architecture: sun4u
Application architecture: sparc
Hardware provider: Sun_Microsystems
Domain: avs
Kernel version: SunOS 5.7 Generic October 1998
```

Graphics Hardware

Tested Platforms

This release of AVS uses two mechanisms to produce its screen renderings of Geometry Viewer objects and scenes: a **software renderer** that implements a set of graphics primitives (polygons, lighting model, perspective, shading, color, etc.) in software, rendering the resulting image into an X Window System image; and a **hardware renderer** that uses the native graphics hardware and the OpenGL graphics library. An additional AVS executable, *avs.xgl* is also included to provide continued XGL graphics library support. **Note:** OpenGL only provides acceleration for the Creator3D and later framebuffers; use *avs.xgl* for older framebuffers for improved performance.

The **software renderer** will work on any supported SPARCstation model with an 8- or 24-plane color frame buffer, the XPutImage *xlib* call, and a local X server that supports them.

You can use the *xdpyinfo* command to list the X server support for your display.

The **hardware renderer** will work on the 24-plane GS, GT, ZX, TZX, SX, Creator3D and Evans and Sutherland Freedom graphics adapters. Note that from SunOS 5.5 onwards the GS and GT device drivers are no longer supported by Sun Microsystems.

See the "Geometry Viewer" section below for specific information on which rendering features are supported by the software and hardware renderers.

This release was developed and tested on Sun SPARCstations equipped with the following color graphics adapter boards:

- GX graphics adapter (8-bit color frame buffer)
- S24 graphics adapter (24-bit color frame buffer)
- TZX graphics adapter (24-bit color frame buffer)
- SX graphics adapter (8-bit PseudoColor and 24-bit TrueColor)
- Creator3D graphics adapter (24-bit color frame buffer)
- Elite graphics adapter

Determining Your Graphics Adapter

To interactively determine which graphics board your SPARCstation has, type the command:

```
/usr/sbin/prtconf -F
```

This prints the device pathname of the console framebuffer. The following are examples (this will depend on the system type and configuration):

```
/sbus@1,f8000000/cgtwelve@1,0        a GS (a.k.a a cgtwelve)
/obio/cgfourteen@1,0                 an SX (a.k.a. a cgfourteen)
/iommu@f,e0000000/sbus@f,e0001000/SUNW,leo@3,0
                                     a ZX (a.k.a. a "LEO")
/SUNW,ffb@1e,0                       a Creator3D (a.k.a. the "FFB")
```

Graphics adapters are described on various *man(4S)* device driver man pages:

```
man cgsix          the GX adapter
man cgtwelve       the GS adapter
man gt             the GT adapter
man leo            the ZX adapter
man cgfourteen     the SX adapter
man tcx            the S24 adapter
man ffb            the Creator3D adapter
```

The absence of any local color graphics display hardware means that, at best, you will only be able to run AVS on your SPARCstation as a remote X client from another color workstation or a color "X terminal".

In all cases, you should read the Configuration Guide that comes with your Sun platform to see if special setup procedures are required.

SX Memory and sxconfig

Note that in order to run the SX system using the TrueColor visual (and have AVS do accelerated hardware rendering on this adapter), the system command */usr/kvm/sxconfig* may have to be run to configure sufficient memory for the SX video system. Please type *man sxconfig* to find out how to determine your system configuration and how to reserve memory for the video system. Note that the machine will have to be rebooted using the *-r* option after running *sxconfig* for the changes to take effect.

Creator3D and ffbconfig

AVS 5 requires that the FFB frame buffer be configured to use gamma-corrected visuals, and that the gamma-corrected visuals are first in the list of visuals supported by the frame buffer. You can use the frame buffer configuration command */usr/sbin/ffbconfig -propt* to confirm that the device is correctly configured.

This command prints the current configuration. For information on what the current options mean, use the *man ffbconfig* command to display the *ffbconfig* man page.

The recommended configuration command is:

```
ffbconfig -res 1280 -deflinear true -linearorder first
```

This configuration sets a 1280x1024x76 video mode and, more importantly, it sets gamma-corrected visuals and makes them first in the X11 visuals list. You will have to be root to reconfigure the device; you should do this only when the window system is not running. You can

use the *-propt* option at any time.

If the device is not correctly configured, the graphics (geometry, images, and so forth) rendering will be too dark; that is, not gamma-corrected.

If you are running the device in stereo mode, you will want to set a different -res option, but with the same -deflinear and -linearorder options. Again, see the output of *man ffbconfig*.

Dialbox and bdconfig

If you wish to use a dialbox with AVS (see the User's Guide for usage) you will need to configure the port into which the device is physically connected. The command to use is */usr/sbin/bdconfig*: type the command *man bdconfig* for details. To enable the device in AVS you may use the *-dials* command line option to the avs command or add a *DialDevice* line to your .avsrc file.

Memory

AVS requires that the SPARCstation upon which AVS will execute be configured with a minimum of 16 megabytes of main memory. More real memory, such as 24 or 32 megabytes, is strongly recommended. Performance will be improved if there is more real memory. In addition to real memory requirements, there are minimum system swap space and shared memory segment size requirements. These are described below.

To see how much memory your system has, enter:

```
/usr/sbin/prtconf
```

"Memory size" is printed within the first few lines of output.

Disk Space

It requires approximately 76 megabytes of disk storage to install AVS from the release media. More disk storage may be required for swap space (see below).

Software Prerequisites

Some of the procedures described in this section require superuser (root) privileges to perform. You may require the assistance of a System or Network Administrator to carry out these procedures.

SunOS 5.7 (Solaris 7)

AVS 5.5 for Sun SPARCstations requires release 5.7 or later of the SunOS operating system (Solaris 7).

A different version of AVS 5 is required for SunOS 5.5 and 5.6 systems. You will need to use AVS 5.4 or contact support@avs.com or your local sales office for further information.

The */usr/bin/showrev* will show you which version of the operating system is running.

Graphics Libraries: OpenGL

AVS 5.5 provides a default AVS executable called *avs* that uses OpenGL for hardware rendering. It provides accelerated graphics support on Creator3D and more recent framebuffers. Older frame buffers are supported by OpenGL but performance will be slower than hardware rendering using the XGL graphics library. AVS 5.5 provides a separate executable, *avs.xgl*, discussed in the next section to deliver XGL hardware rendering.

You can always determine which version of AVS 5.5 you are running using the "-version" command line option. The graphics library is listed at the end of the version string.

The OpenGL runtime environment is required for both hardware and software rendering. If you do not already have OpenGL installed on your system, you will need to download and install the runtime environment from Sun Microsystems. The web site address is

```
http://www.sun.com/solaris/opengl/download
```

The installation will require that you stop and restart the window system in order to load the required X GLX extension support OpenGL.

To see if the XGL runtime environment is present, use the */usr/bin/pkginfo* utility. Look for the following lines:

```
application SUNWglrt       Solaris OpenGL Runtime Libraries
application SUNWglrtu      Solaris OpenGL Platform Specific Runtime Libraries
```

You can also check to see if the library */usr/openwin/lib/libGL.so.1* exists. The AVS executable *avs* expects to find the OpenGL runtime library in this location.

If it is not, you will need to set the LD_LIBRARY_PATH environment variable to include its non-standard directory.

You can use the *ldd* command to determine which shared object libraries AVS requires, whether they can be found, and the library locations for those libraries found.

Graphics Libraries: XGL

AVS 5.5 provides an optional AVS executable called *avs.xgl* that requires the presence of version 3.x of the Sun XGL graphics library in order to initialize. If you normally use the XGL executable, you may want to make it your default version of AVS as follows:

```
cd $AVS_PATH/bin
mv avs avs.ogl
ln -s avs.xgl avs
```

You can always determine which version of AVS you are running using the "-version" command line option. The graphics library is listed at the end of the version string.

The XGL runtime environment is required for both hardware *and* software rendering when using the *avs.xgl* secondary kernel. XGL is shipped bundled on the main SunOS CD, and should be installed automatically as part of the main OS installation.

To see if the XGL runtime environment is present, use the */usr/bin/pkginfo* utility. Look for the following lines:

```
application SUNWxgldg      XGL Generic Loadable Libraries
application SUNWxgler      XGL English Localization
application SUNWxglft      XGL Stroke Fonts
application SUNWxglrt      XGL Runtime Environment
```

You can also check to see if the library */opt/SUNWits/Graphics-sw/xgl-3.0/lib/libxgl.so.3* exists. The AVS executable *avs.xgl* expects to find the XGL runtime library in this location.

If it is not, you will need to set the LD_LIBRARY_PATH environment variable to include its non-standard directory.

You can use the *ldd* command to determine which shared object libraries AVS requires, whether they can be found, and the library locations for those libraries found.

Window System

Sun OpenWindows 3.x

AVS runs as an X Window System client. AVS 5 was implemented and tested under Sun OpenWindow and the Common Desktop Environment (CDE).

Other X Window Systems

AVS may also be able to execute under the MIT X Window System Version 11 Release 6 Sun server (color) distributed on the MIT X Window System tape. However, Advanced Visual Systems has not tested AVS under this configuration.

The server must have been built to be aware of the particular color graphics adapter in your SPARCstation (cgsix0, cgtwelve, gt0, etc.) If it was not, you will not be able to start the X server.

Openwin Libraries: can't find libXext.so Error Messages

AVS expects to find the X11 libraries in their default installed location (*/usr/openwin/lib*).

If they are not, then you will need to set the LD_LIBRARY_PATH environment variable to include their non-standard path in order for AVS to run.

You can use the *ldd* command to determine which shared object libraries AVS requires, whether they can be found, and the library locations of libraries found.

Swap Space: Increasing Limits

Recommended Swap Space Size

Most SPARCstations come with the default system swap space size set to about twice the amount of real memory.

We recommend increasing the swap space available to your system to at least 64 megabytes, or twice the amount of available memory, whichever is *larger*. Very large datasets flowing through networks with many modules and connections may require even larger swap spaces. Some modules—such as the **dot surface** module that renders data values as spheres, and the **compute gradient** module that produces 12 bytes for every input value— require very large output areas.

People with very large real memory sizes and smaller datasets may require less than 64 megabytes.

Determining How Much Swap Space You Have

To see how much swap space is available to your system, type the following command:

```
/usr/sbin/swap -s
total: 9088k allocated + 1500k reserved = 10588k used, 32724k available
```

Add the last two figures together to find the total swap space available on the system, both in the system swap space partition and any additional swap space that might have been defined for the system on other partitions. In this case, the system has approximately 43 megabytes of swap space (10588k + 32724k).

Running Out of Swap Space: Abort or Hang

When AVS runs out of swap space, it aborts or hangs. In many cases, AVS itself receives the error during a *malloc* call. In this instance, either individual modules or the entire AVS system will abort with the following error message:

```
Failure allocating memory:
See Installation/Release Notes to increase swap space and/or
shared memory segment size
```

This message may be followed by many additional "tcp read" error messages as each module expires. Scroll to the top of the error messages to see this first message that indicates the actual cause of the failure.

In some cases, other parts of the system will generate the error and you will not see the above message.

Monitoring Swap Space and Shared Memory Segment Use

If you suspect that swap space may be the reason AVS is dying, you can restart AVS, repeat the sequence of steps that led to the abort or hang, while monitoring AVS's consumption of swap space by entering successive */usr/sbin/swap -s* commands. The *ipcs -a* command, if entered as root, will also show you the amount of shared memory in use (SEGSZ column).

How to Increase System Swap Space

To increase the system swap space size, see the Sun document *SunOS 5.x Routine System Administration Guide* in the chapter titled "Configuring Additional Swap Space." This document is available

online through the Sun Answerbook facility. Also, typing *man swap* and reading the *-a* option gives useful information.

The procedure involves optionally using the */usr/sbin/mkfile* command to create a swap space file, then using the */usr/sbin/swap -a* command to add the swap space. To make the increase in swap space permanent rather than just for the current session, you must also edit the */etc/vfstab* file. Again, see the "Configuring Additional Swap Space" chapter in the *Routine System Administration Guide* and the *swap* man page for more information.

If you are supporting diskless workstations on a central server, you may need to increase the swap space available to each client workstation that will run AVS.

Shared Memory Segment Size

AVS makes heavy use of shared memory to improve performance. Shared memory reduces the amount of memory AVS needs, and makes AVS run faster because less data must be copied between modules.

Shared Memory Problems: shmget failed Error Messages

If shared memory is not enabled on your system, or if the amount of shared memory AVS requests is larger than your system limit, or if the number of shared memory segments requested is larger than your system limit, you will see a message like the following:

```
shmget failed: invalid argument
```

AVS then falls back on making individual copies of the datasets in each module and the kernel instead of using shared memory, consuming substantially more real memory, swap space, and time copying the data between modules and the AVS kernel.

Finding Out If Shared Memory Is Enabled and Increasing Your System Limits

As SunOS 5.7 comes "out of the box," there are several shared memory parameters that may not be configured correctly:

- Shared memory may be altogether disabled (this is often the default).
- Any one of these limits may be too small:

shmmax

This is the maximum shared memory segment size (in bytes) that a process can acquire in a single request. The documentation states that the default is 131072 bytes. This is far too small for AVS.

For example, a 512x512 image requires 1 megabyte to store. A 40x32x32 5-vector single precision floating point curvilinear dataset with accompanying points information requires around 1.3 megabytes. Many datasets are much larger. AVS modules usually request at least as much memory as the size of the input dataset, and often considerably more. **compute gradient**, for example, needs 12 output bytes for every input data value. This will result in shared memory failing for large datasets.

shmseg

This is the maximum number of shared memory segments that a single process can have. The documentation states that the default is 6. This is, again, far too small for AVS.

To see if shared memory is turned on, type this command:

```
ipcs -a
```

If you see a message, "Shared memory facility not in system...", then shared memory is either not turned on or has not been used by the system. You turn it on by establishing the limits in */etc/system* described in the next paragraph.

If the **shmseg** is less than about 30, and the **shmmax** size is less than at least 10 megabytes, you should raise these limits. People with very large, complex networks may need a **shmseg** limit greater than 30. People with larger datasets, and who use modules with large output sizes will need a larger **shmmax** size. (There is no apparent disadvantage to setting limits extremely high.)

To enable shared memory and/or modify its parameters, edit the */etc/system* file as root:

- Add lines like the following (or edit the existing lines to have these values):

```
set shmsys:shminfo_shmmax=65536000
set shmsys:shminfo_shmmin=1
set shmsys:shminfo_shmmni=100
set shmsys:shminfo_shmseg=32
```

- Reboot the system to have the changes take effect.

Using AVS on Sun SPARCstations

The following sections describe in detail the differences between using AVS on a SPARC workstation and AVS as it is described in the standard documentation.

Peripherals

Dial Box

The Dial Box I/O device is supported in SunOS 5.7.

Spaceball

The Spaceball I/O device mentioned in the *AVS User's Guide* is supported in this release of the product.

Stereo Support

Both versions of AVS 5.5 (OpenGL and XGL) include basic support for stereo viewing using CrystalEyes stereo goggles from StereoGraphics. See the "AVS 5.5 on UNIX Platforms" chapter for more general information.

Stereo needs to be enabled before the X server is started up; if stereo is not enabled you would need to do the following:

- Exit from the X server using the CDE "Exit" button and return to the main login prompt.
- Under the Options menu, select "Command Line Login"; hit Enter or Return to get a new login prompt and login as root.
- Type "ffbconfig -res stereo try" to attempt to switch into stereo. This will switch into a possibly lower resolution mode that will support stereo. The text on the screen may be scrambled but this doesn't seem to be a problem once the X server restarts. If you get messages about this mode not being supported for your monitor, etc. then please consult your system documentation for further information.
- Exit and hit the "Enter" button to restart the X server and log in again.
- If the screen is a smaller resolution, type "avs -size 900x700" to shrink AVS to better fit the new screen size. OpenGL stereo should now work.

- To return to the original resolution, repeat the steps above but type the following instead:

  ```
  ffbconfig -res 1280 try
  ```

Window Manager

AVS works with any of the standard X Window System window managers, including the Common Desktop Environment and Motif. No user modifications are necessary in order to use AVS with these window managers but there are some modifications that you might want to make as a matter of personal preference. See the "Window Manager" section in Chapter 3, *AVS 5.5 on UNIX Workstations* for more information on modifying how AVS behaves under Motif and OPEN LOOK.

Starting AVS

The issues here that differ from the account in the AVS *User's Guide* and the *avs* man page are: scaling the AVS interface to fit your display resolution, and setting the correct X server VisualType for your graphics adapter.

All involve making changes to your personal AVS startup file (*.avsrc*) in your HOME directory. If you don't already have a *.avsrc* file, copy the sample file from <*install_dir*>*/avs/runtime/avsrc* to your home directory and edit it to include the necessary lines. <*install_dir*> is replaced with the directory in which AVS is installed.

Display Size: GX, GS, SX, S24

This affects GX, GS, SX, and S24 graphics adapters.

AVS normally expects to run on 1280 x 1024 resolution monitors, and sizes its interface accordingly. The display usually supplied with GX and GS graphics adapters is 1152 x 900. Add the following lines to your personal *.avsrc* file or use the *-size* command line option:

```
NetworkWindow          850x850+250+20
ScreenSize             1080x864
```

to size AVS reasonably for your monitor. ZX, Creator3D, Evans and Sutherland Freedom, GT and GXplus adapters have 1280 x 1024 monitors.

The AVS "-size" command line option provides the same functionality that the ScreenSize startup option does and can be used to experiment with the best screen size.

Visual Type: Pseudocolor on Truecolor Adapters

This affects GS, GT, ZX, SX, Creator3D and Freedom graphics adapters. It may also affect other 24-plane graphics adapters that you may have, such as the S24.

No matter which graphics adapter is present in your SPARCstation, the X server on SPARCstations always has a PseudoColor visual (as reported by the *xdpyinfo* command) as its default visual. By default, AVS always uses the default X visual when it starts.

On GX graphics adapters, this is fine. AVS will use the correct PseudoColor visual for this 8-bit platform.

However, on the graphics adapters mentioned above, AVS will also come up using the 8-bit default PseudoColor visual. To force AVS to use the 24-bit TrueColor visual that these adapters are capable of supporting, add this line to your *.avsrc* file:

```
VisualType      TrueColor
```

You may also want to use the new *-vistype* command line option documented in the "Unexposed command line options" section of the UNIX chapter. For information on checking and changing your default visual, see Chapter 3, *AVS5.5 on UNIX Platforms*.

Changing the OpenWindows Default Visual Permanently

You can also force OpenWindows to use a particular X Window System default visual when it initializes.

In your sequence of startup files, find the line that starts OpenWindows. For a "vanilla" user that starts OpenWindows when they login, this is usually the line "*/usr/openwin/bin/openwin*" in the user's *.login* file.

Change this line to read:

```
/usr/openwin/bin/openwin -dev /dev/graphicsadapter defdepth 24
```

where *graphicsadapter* is the id of the graphics adapter in your workstation, as described in the "Graphics Hardware" section above. This practice should not, however, be necessary and is not recommended.

Loader Messages at Startup

AVS 5 was dynamically linked against certain system libraries. These libraries must be present for AVS to start, if they are not you will see a message of the form

```
ld.so.1:  avs:  can't find libGL.so.1
```

or if you are using *avs.xgl*, a message like this:

```
ld.so.1:  avs:  can't find libxgl.so.3
```

See the "Graphics Libraries" sections above for information on how to proceed.

Image Viewer

All functions and behavior in the Image Viewer are as described in the AVS *User's Guide*. The Image Viewer will use 8-bits of color on GX graphics adapters, and 24-bits on graphics adapters using the TrueColor visual.

On pseudocolor systems, the Image Viewer's **Images** submenu will include a choice of dithering options.

Network Editor

The Network Editor functions as described in the AVS *User's Guide* and *Module Reference* manual, with these few exceptions.

Modules

The following modules documented in the *AVS Module Reference* manual do not appear in the AVS release for Sun SPARCstations because they require rendering features that are not supported in this release:

alpha blend (requires hardware-assisted alpha transparency)
transform pixmap (requires 2D texture mapping)

Graph Viewer

Other than the fact that image data loaded as a background to a plot on a GX graphics adapter will be dithered to 8-bit pseudocolor rather than appear in true color, there are no differences between AVS Graph Viewer behavior on SPARCstations and the Graph Viewer descriptions in the AVS documentation.

Geometry Viewer

The Geometry Viewer (or the **geometry viewer** module) will be using either the software renderer or the hardware renderer to produce the contents of its scene windows.

OpenGL differences

The default AVS executable *avs* provides hardware rendering using OpenGL does not support several Geometry Viewer controls that were custom implemented for XGL renderer features. The following controls will not appear in the OpenGL executable:

- Normal Mapping
- VLTM
- Adaptive Subdivision
- Nsegs Subdivision
- Combined Subdivision
- Global Anti-Aliasing

Color and Single/Double Buffering

On the 24-plane GS system, XGL uses a 12-plane true color visual. Thus, when you are in the default double buffer mode, you are seeing only 12 planes of color (4096 possible colors), rather than the full 24 planes (16,777,216 possible colors). To see a full 24-plane true color rendering, you must change to single buffer mode by switching off the **Double Buffer** button on the **Cameras** submenu.

There is, at present, no AVS startup option that will automatically switch off double buffering.

"Double buffer" means that XGL renders its output image into invisible planes, then makes those planes visible, switching back and forth between two sets of planes. "Single buffer" means that XGL renders its output image directly to visible planes. If single buffer mode is set, you can watch the object being drawn; if double buffer mode is set, the completed picture simply appears in the window.

OpenGL does not provide support for the GS and GT graphics adapters. See *http://www.tgs.com* for alternate OpenGL drivers for boards not supported by Sun.

Rendering Features

The following table lists the rendering features described in the "Geometry Viewer" chapter of the *AVS User's Guide*, including AVS 5 features down the left column, and the AVS software renderer and the AVS hardware renderer's XGL implementation across the top. The table intersections show which features are present on each platform, and draw your attention to more detailed explanations of behavior later in this section.

Table 8-1. Geometry Viewer Behavior Across Platforms

Rendering Feature	Software Renderer	Hardware (XGL)	Hardware (OpenGL)
Arbitrary Clip Planes	yes, 8 planes (note 1)	yes	no
Geometric			
Volume Rendering	yes (note 2)	no	no
Vertex Transparency	no (note 3)	no	no
Vertex Colors	yes (note 4)	yes	yes
Edit Property			
RGB/HSV Colors	yes (note 4)	yes	yes
Ambient	yes	yes	yes
Diffuse Lighting	yes	yes	yes
Specular Highlights	yes	yes	yes
Gloss	yes	yes	yes
Transparency	yes (note 5)	yes/no (note 7)	no
Metallic	yes	yes	yes
Edit Texture	yes	yes	yes
2D Texture Mapping	yes	yes (note 8)	yes
3D Texture Mapping	yes	no	no
Filtered Textures	no	yes	yes
Tile Textures	no	yes	no
Alpha Textures	yes	yes	yes
Rendering Options			
Points	yes	yes	yes
Lines	yes	yes	yes
Smooth Lines	no	yes/no (note 9)	no
No Lighting	yes	yes	yes
Flat Shading	yes	yes	yes
Gouraud Shading	yes	yes	yes
Outline Gouraud	yes	yes	yes
Phong Shading	no	no	no
Backface Properties			
Cull front	no	yes	yes
Cull back	yes	yes	yes
Flip normals	no	no	no
Lights			
Number of Sources	16	8/16 (note 11)	8
Ambient	yes	yes	yes
Directional	yes	yes	yes
Bi-Directional	yes	yes	yes
Point	no	yes	yes
Spot	no	yes	no
Light Colors	yes	yes	yes

Table 8-2. Geometry Viewer Behavior Across Platforms (continued)

Rendering Feature	Software Renderer	Hardware (XGL)	Hardware (OpenGL)
Cameras			
Depth Cue	yes	yes	yes
Global Anti-Alias	no	yes/no (note 13)	no
Accelerate	no	no	no
Perspective	yes	yes	yes
Axes for Scene	yes	yes	yes
Front/Back Clipping	yes	yes	yes
Sorted Transparency	no	no	no
Polygonal Spheres	yes (note 6)	no (note 10)	no
Double Buffer	no	yes/no (note 12)	yes
Stereo	no	yes (note 13)	yes (note 13)
Labels			
Drop Shadow	no	no	no
Title	yes	yes	yes
Stroke	no	no	no
Kanji	yes	no	no

Notes

1. Arbitrary clipping planes are used by the **clip geom** module.

2. "Volume Rendering" is here defined in a very narrow sense: support of a specific option to the **GEOMedit_texture_options** call that is used by the **volume render** module.

3. Vertex transparency is used by the **colorize geom** module.

4. When using the software renderer, the number of colors that will appear on the screen (216 pseudocolor, 16,777,216 true color, or some number in between) is dependent upon the X server visual support present on the *display* hardware. Internally, 24-bit true color is always used and will be output on the **geometry viewer** module's image output port.

5. The software renderer supports two-pass transparency. Two-pass transparency correctly renders a transparent surface that is over an opaque surface. (Multi-pass transparency is required to correctly render multiple overlapping transparent surfaces.)

6. The software renderer is able to render spheres directly instead of subdividing them into polygons. This saves a tremendous amount of memory when rendering spheres. The algorithm's

execution speed is bound by the size of the spheres it has to render. This feature may be disabled by toggling the **Polygonal Spheres** button under the Cameras menu. Spheres will now be rendered by subdividing them into polygons.

7. The XGL hardware renderer supports two-pass transparency on the ZX, SX and GT adapters, but not on the GS. The SX does this in software so you can expect some performance degradation on that system when there are transparent objects in the view.

8. XGL currently does all texture mapping in software and there is a considerable performance degradation when using this feature, particularly with large (greater than 256x256) texture images. There are also issues regarding texturing quality on non-planar surfaces. Sun Microsystems is addressing these issues. Please call AVS Customer Support for the latest information if this is of concern to you.

9. The XGL hardware renderer supports antialiased lines on the ZX, SX and GT adapters, but not on the GS.

10. Spheres are subdivided into polygons when using the XGL based hardware renderer.

11. The GS adapter supports 8 light sources, all others support 16.

12. The GS and S24 adapters do 12 bit dithered double buffering. Toggling the **Double Buffer** button *off* will render geometry in 24 bits non-dithered. The double buffer button will be absent on all other adapters since they render at full 24-bit resolution when double buffered.

13. Supported on the (T)ZX and Creator3D adapters. You will need to configure the frame buffer device for stereo viewing, see the *leoconfig* and *ffbconfig* commands, respectively.

Programming Considerations

AVS, it's modules, and component libraries were compiled using the ANSI C SPARCompiler *cc* version SC4.2. Executables were linked using libraries located in */opt/SUNWspro/lib*. FORTRAN components were compiled using the FORTRAN *f77* SPARCompiler version SC4.2. Executables were **dynamically** linked against system libraries.

Compiling and Linking Modules

You should use the make file in <*install_dir*>*/avs/examples/Makefile* as the template for your own FORTRAN and C module make files. (<*install_dir*> is the directory in which AVS is installed. This make file in turn includes the file <*install_dir*>*/avs/include/Makeinclude* that contains additional macro definitions appropriate to the Sun Solaris platform.

Static Versus Dynamic Linking

AVS 5 was linked dynamically against system libraries. The STATICFLAG symbol in <*install_dir*>*/avs/include/Makeinclude* is set to NULL, making dynamic linking the default when you compile and link modules.

The module user may see error messages stating that the system cannot locate certain libraries. In this case the LD_LIBRARY_PATH environment variable can be modified to include the path of the versions of the missing shared object libraries.

AVS on SunOS 5.7: Known Problems

This section lists the known problems with this release of AVS 5.5 that are unique to SunOS 5.7 on SPARCstation platforms.

X8689: SUN ZX, GT: Layout Editor widget outline invisible

Problem Description:

When moving a widget from the main panel onto a page, the widget outline is invisible when it is displayed in windows of depth 24 (which includes all AVS windows). This is a problem when using the TrueColor visual for AVS.

Workaround: A partial workaround has been implemented within AVS. When using the TrueColor visual (on the ZX and GT only) widget outlines are drawn using the parent window visual instead of the 8 bit deep RootWindow. This means the outline will be visible only while in its parent window. So to place a widget within a page, for example, move the widget to the page and release the mouse button so that the widget is reparented to the page. Now move the widget again and its outline will be visible while in the page.

X10615: SX: AVS logo is random when using TrueColor Visual

Problem Description:

The AVS logo is usually not drawn correctly on the SX when

using the TrueColor visual.

Problem: Label color affected by Perspective and Depth Cueing

Problem Description:

Use of Perspective and Depth Cueing in a view may affect the color of labels in the same view.

AVS on SunOS 5.x: Fixed Problems

This is the list of fixed problems for AVS 5.5 on the SunOS 5.7 on SPARCstation platforms.

17582: Elite 3D graphics adapter support

Problem Description:

Support has been added for the Elite board. Customers may require the following patches to work properly: 105363-06, 105361-04 and for OpenGL support, 106022-02.