Subject: Re: IDLgrVolume
Posted by Karl Schultz on Tue, 21 Jan 2003 23:07:05 GMT
View Forum Message <> Reply to Message

"Lyubomir Zagorchev" <lzagorch@cs.wright.edu> wrote in message
news:b0k0vs$cvf$1@proxy1.wright.edu...

>
> First, thank you for your replies.
>
> The hardware rendering is on and I am running the program under Windows
> 2000,
> on Pentium 4, 2.0 GHz dual processor, 512 Ram, with Nvidia GeForce3
graphics
> card.
>
> I'll make the program and a dummy dataset available for independent
testing
> asap,
> but in the meantime is there any special hardware that can speed up the
> rendering
> of volumes (like particular graphics card, etc.)?

IDLgrVolume::Draw is implemented totally in software, so Rick's advice on getting the fastest CPU and the most memory you can is good advice. Since your machine is dual processor, you are also ahead of the game, as IDLgrVolume was multi-threaded even before some other IDL functions received MT support. You might want to bring up your Task Manager with the CPU performance graphs and see if you observe both processors getting hammered during a long volume rendering.

Selecting IDL hardware vs software rendering won't make a significant difference because IDL is just drawing a 2D image to the screen, and the time needed for that is tiny compared to the rendering.

There's another way to display a volume by building a series of slices with polygons, texture mapped with the appropriate data and alpha information. It provides a really decent approximation to the volume and displays pretty quickly. One problem with it is that you have to render the slices back to front since blending is involved, so you have to change your rendering order based on the orientation to the view point. I'm pretty sure I've posted about this before, but we can revisit if there is interest.

Hardware:

There are some OpenGL accelerator cards that support 3D texture mapping. 3D texture mapping can be used to display volumes, but the speed depends on how the card vendor implemented the 3D texture mapping support. A long time ago, RSI experimented with a card from HP that could do this with hardware

acceleration, but I don't remember the outcome. I know part of the issue
was that the OpenGL interface for 3D texture mapping was still under
significant revision and there was much peril in coding anything using these
interfaces. Today, there are not very many cards that support the 3D
texture API. So, currently, IDL doesn't try to use 3D textures for volume
rendering. If the OpenGL driver does this in software, I think that the
technique is similar to the slice idea I mentioned above.

Some companies make plug-in volume rendering accelerator cards and the
software that goes with them. I think Mitsubishi is one of them
(http://www.mitsubishielectric.com/news/1999/052499.htm), but there are
probably others. I suppose that if all the right things were in place, such
a product might offer a C API to the card and you could write an IDL DLM to
send a volume to the card, pull the resulting image back, and draw it to the
screen in IDL.   But cards like these are expensive and often have upper
limits on volume size that are not large enough for some people.

Other software:

There are a bunch of free and not-so-free volume rendering software packages
out there if all you want to do is look at volumes and do whatever else they
support. Some may support volume accelerators like the Mitsubishi, but most
are going to be software implementations. They may offer some features
and/or algorithm selection that provide some choices in performance ranges.

Hope this helps,
Karl