Subject: Re: IDLgrWindow, IDLgrVolume and alpha channel
Posted by Rick Towler on Fri, 24 Jan 2003 20:08:03 GMT
View Forum Message <> Reply to Message

"Sebastian" wrote in message

> I am rendering an IDLgrVolume into a IDLgrWindow. When I get the image
> data from the Window, the image has only 3 channels. How can I get the
> alpha channel of this image? The alpha channel information surely must be
> somewhere, because volume rendering needs it.

<speculation>

I'm sure Karl can give you a technical answer but the short of it is that I am pretty sure you can't get the alpha channel from a window by using the Read method. I would guess that by the time the data is written to your display buffer the alpha information is long gone. I don't know if this is a "limitation" of OpenGL, the OS, or IDL and while I can think of a few reasons not to do this, I can't think of a good reason to do it.

</speculation>

You may be able to get the data you need in some other way. Maybe if you post some specifics on what you are doing someone might come up with a creative alternative.

-Rick

Subject: Re: IDLgrWindow, IDLgrVolume and alpha channel
Posted by Karl Schultz on Sat, 25 Jan 2003 00:06:07 GMT
View Forum Message <> Reply to Message

"Rick Towler" <rtowler@u.washington.edu> wrote in message
news:b0s6ln$1k30$1@nntp6.u.washington.edu...
>
> "Sebastian" wrote in message
>
>> I am rendering an IDLgrVolume into a IDLgrWindow. When I get the image
>> data from the Window, the image has only 3 channels. How can I get the
>> alpha channel of this image? The alpha channel information surely must
be
>> somewhere, because volume rendering needs it.
>
> <speculation>
>
> I'm sure Karl can give you a technical answer but the short of it is that
I

> am pretty sure you can't get the alpha channel from a window by using the
> Read method. I would guess that by the time the data is written to your
> display buffer the alpha information is long gone. I don't know if this
is
> a "limitation" of OpenGL, the OS, or IDL and while I can think of a few
> reasons not to do this, I can't think of a good reason to do it.
>
> </speculation>

Right.

The only way to get alpha information back out of the frame buffer is if the graphics system supports "destination alpha". This means that there are some planes in the frame buffer dedicated for alpha planes. Destination alpha is primarily used for some advanced rendering techniques such as compositing. There aren't very many devices that support destination alpha, and there is currently no way to leverage it with IDL. You'll find destination alpha more often on really high-end workstations.

The IDL volume renderer generates an RGBA IDLgrImage and draws that to the screen. There is no way to gain access to this image, and I'm not sure that would make sense, because the contents of the image would depend on on the current View, etc. Besides, it get created and destroyed during the draw anyway. I suppose that a new method that is similar to Draw could be created that would return an IDLgrImage (like IDLgrWindow::Read), but that does not exist today.

> You may be able to get the data you need in some other way. Maybe if you
> post some specifics on what you are doing someone might come up with a
> creative alternative.

Yes, knowing some more specifics might help. But here are a couple of ideas:

1) Try rendering the volume with a greyscale ramp and read it back. The same data will be in all three channels. Although this is really color data, it might be close enough to the desired transparency values. I can look at the validity of this approach more closely if there is interest.

2) If you think that you need the alpha to coordinate with drawing other objects, like an isosurface, take a look at the ZBUFFER keyword on IDLgrVolume.

Hope this helps,
Karl