"Karl Schultz" wrote >

> "Sebastian" wrote

>> Is there a difference between using a IDLgrBuffer and IDLgrWindow for
this
>> problem?
>
> No. Both IDLgrBuffer::Read and IDLgrWindow::Read return RGB arrays. I
> suppose that IDLgrBuffer could be fixed up to use destination alpha and
> return RGBA with Read, but I think that getting the data from IDLgrVolume
> addresses your need more directly. Remember that the destination alpha
> values you would pull out of the frame buffer may not correspond directly
to
> the source alpha values of your primitive (volume). They might, if the
> blend equations were just right and there were no other primitives in the
> scene. Anyway, my point is that trying to derive the original alpha
values
> by getting them from the device frame buffer is a roundabout way of doing
> it.

If we're throwing around feature requests I want to get in!

I think that fixing IDLgrBuffer to use destination alpha is a good idea.
Recently I have been procedurally creating textures (rendering to a buffer
then texmapping) and although I haven't had a need for the alpha channel
yet, I can see the possibilities.

But I wouldn't want this to get in the way of that new renderer you are
working on. You know, the one that handles order-independent transparency?
:)

-Rick