Subject: Re: pixmap drawables in Object Graphics?
Posted by David Fanning on Tue, 18 Dec 2001 00:54:34 GMT
View Forum Message <> Reply to Message

Martin Downing (martin.downing@ntlworld.com) writes:

> This is similar to previous queries on object graphics and pixmaps but I
> would appreciate running this by the experts.
> I am writing a program to fit projections of a 3d surface model to its
> silhouette in an image (e.g. a radiograph). This method allows an estimate
> of object position to be recovered from the knowledge of the object shape
> and the image. My 3d data is a triangulated mesh which can be best stored as
> a IDLgrPolygon object. This is attractive as you can then easily specify a
> graphics model to render the object at specific rotations, and projections
> of complicated polygon objects can then be drawn rapidly using OpenGL.
>
> However, as this is part of a fitting process, I then read the drawable back
> into an image buffer using say tvrd(), do some image processing to get a
> goodness of fit quantity and repeat until a sufficient fit is found.

Well, I don't think you are going to be doing
any TVRDing in object graphics windows. :-)

The TVRD equivalent in object graphics is probably
the READ method on a window object, but that returns
an image object (with 24-bit image data). Not the sort
of thing you will be doing a lot of image processing
on, probably.

> I do
> not need to see each projection in an exposed draw widget, but as far as I
> can gather, pixmaps are not implemented in object graphics.

The IDLgrBuffer object is the object graphics equivalent
of a pixmap. But, again, this is no 2D graphics window of
the sort you seem to expect. *Everything* in object graphics
is 3D. The object graphics system *is* a 3D system. That is
the point of it.

> So as I see it,
> my only option using object graphics is to use normal draw widgets, which
> seems like overkill.

I'm not sure I understand this statement. Object graphics
and "normal" draw widgets are mutually exclusive. You can
use one or the other, not both. Some things (rendering
complex polygons come to mind) are perfect for object
graphics. Other things (say, working with 2D images) often

work better in direct graphics windows. Which you use depends entirely on what makes sense.

Here, it might make sense to have overlapping widget hierarchies with both object and normal draw widgets mapped into the same real estate in your GUI. Then, depending upon what you wish to display, you can choose one window or the other.

Come to think of it, that might be a good excuse to write a combination window compound widget. Of course, it should be written as an object. :-)

Cheers,

David

--
David W. Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155
