Subject: Object graphics questions and comment
Posted by Paul van Delst on Fri, 08 Aug 1997 07:00:00 GMT
View Forum Message <> Reply to Message

Hello,

I'm in the process of writing a viewer for atmospheric absorption spectra - the data files of which often contain 1000's of spectra. I decided to use IDL object graphics for a number of reasons:

- makes it easy to do zoom boxes
- makes it easy to do overplotting on resizeable window (sort of like David Fannings XWINDOW but without the need for a wrapper)
- makes it easy to add annotation
- in the words of Mr. Fanning, makes me look like an "up-to-date and with-it IDL programmer".

Anyway, I create a zoom box on a left button down event, add it to my plot_model and draw the resulting view (info_ptr is a pointer to the info structure ala D.F. and is stored in the user value of the top level base) :

```
  (*info_ptr).zbox = obj_new( 'IDLgrPolygon',
$
(*info_ptr).zbox_norm_xcoords, $
                        (*info_ptr).zbox_norm_ycoords, $
                        color = [ 0, 255, 0 ], style = 1 )
  (*info_ptr).plot_model -> add, (*info_ptr).zbox
  (*info_ptr).window -> draw, (*info_ptr).view
```

For motion events I do the following after updating the dynamic corner of the zoombox:

```
  old_zbox = (*info_ptr).zbox
  (*info_ptr).plot_model -> remove, (*info_ptr).zbox
  obj_destroy, old_zbox

  (*info_ptr).zbox = obj_new( 'IDLgrPolygon', $
                        (*info_ptr).zbox_norm_xcoords, $
                        (*info_ptr).zbox_norm_ycoords, $
                        color = [ 0, 255, 0 ], style = 1 )
  (*info_ptr).plot_model -> add, (*info_ptr).zbox
  (*info_ptr).window -> draw, (*info_ptr).view
```

Now all this works great. I get a lovely zoom box except IT IS SOOOOO SLOW? I test my code on a 1000 element data set and the zoom box can't keep up with the mouse motion events. Increase the number of points and it's unusable.

MY QUESTIONS: Is the snail pace response implicit of IDLs of object graphics (another e.g. IDLs Insight. Good lord) or is there another way to do this without having to do a
(*info_ptr).window -> draw, (*info_ptr).view
every time? In Direct Graphics you simply erase ONLY the box and then redraw ONLY the new box, you don't redraw the entire view. Can this be done using Object Graphics? The documentation is obtuse and unhelpful.

MY COMMENT: IMO, IDLs Object Graphics is necessary for easier programming of user friendly data visualisation tools. However, the speed of Object Graphics verges on being laughable (my perhaps poorly programmed example above notwithstanding). It is a step backwards and does not encourage the average IDL programmer (me) to utilise this powerful tool. My (hopeful) assumption is that future incremental releases of IDLv5 will contain Object Graphics that produce graphics as fast or faster than Direct Graphics.

Phew!

I'm curious - who out there are actually using IDLs Object Graphics capability?

regards,

Paul van Delst
Space Science and Engineering Center
University of Wisconsin-Madison
(608) 265-5357

Subject: Re: object graphics question
Posted by David Fanning on Wed, 04 Feb 2004 14:08:01 GMT
View Forum Message <> Reply to Message

Karsten Rodenacker writes:

> I have a problem with coordinated transformations of IDLgrImage and
> other objects, e.g. IDLgrContour.
>
> Does anybody know why rotating behaves like it does?
> In the small example:
>
> img=read_png(filepath('mineral.png', SUBDIR=['examples','data']))
> m=obj_new('IDLgrModel')
> m->add,obj_new('IDLgrImage',img)
> m->add,obj_new('IDLgrContour',img,color=[255,0,0])
> xobjview,m,/block

>
> rotation shows a quite bizar behaviour.
>
> I would expect that the image moves like one level of the contour object.

You will have to put your image on a polygon as
a texture map to get it to rotate. Images
have *never* rotated. I don't know why.

Cheers,

David
--
David W. Fanning, Ph.D.
Fanning Software Consulting, Inc.
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

Subject: Re: object graphics question
Posted by btt on Wed, 04 Feb 2004 14:54:03 GMT
View Forum Message <> Reply to Message

Karsten Rodenacker wrote:
>
> I have a problem with coordinated transformations of IDLgrImage and
> other objects, e.g. IDLgrContour.
>
> Does anybody know why rotating behaves like it does?
> In the small example:
>
> img=read_png(filepath('mineral.png', SUBDIR=['examples','data']))
> m=obj_new('IDLgrModel')
> m->add,obj_new('IDLgrImage',img)
> m->add,obj_new('IDLgrContour',img,color=[255,0,0])
> xobjview,m,/block
>
> rotation shows a quite bizar behaviour.
>
> I would expect that the image moves like one level of the contour object.
>

Hi Karsten,

Yes, that is wierd... but there is a solution.

You should really be rendering a polygon with the image as a texture

map. IDLgrImage are always displayed at Z=0 (or some such place.) To give the image a real coordinate system you have to attach it to some spatial object, like a polygon or a surface.

```
img=read_png( filepath('mineral.png', SUBDIR=['examples','data']))
m=obj_new('IDLgrModel')
oimg = obj_new('IDLgrImage',img)
oimg -> GetProperty, dim = dim
poly = obj_new("IDLgrPolygon", [0,0,dim[0]-1, dim[0]-1], $
 [0,dim[1]-1,dim[1]-1, 0], $
 color = [255,255,255], $
 Texture_Map = oImg,$
 Texture_Coord = [[0,0],[0,1], [1,1], [1,0]])
m->add, poly
m->add,obj_new('IDLgrContour',img,color=[255,0,0])
xobjview,m,/block
```

Cheers,
Ben

Subject: Re: Object graphics questions
Posted by Michael Wallace on Sat, 12 Mar 2005 00:54:37 GMT
View Forum Message <> Reply to Message

> (3) Is it a hard-and-fast rule that if I am viewing a 240x250 image, I
> just have to set the viewplane rectangle to [0,0,240,250]? I have done
> that but the results don't look right.

No. The VIEWPLANE_RECT is not related to the size of an image. The coordinates are with respect to normalized space. If you do not do any normalization, the viewplane will need to be sized according to the data space (i.e. [xmin, ymin, xmax - xmin, ymax - ymin] for a VIEWPLANE that fits your plot exactly).

Typically, I will instead calculate the scaling factors for converting the X and Y data ranges into normalized space. These scaling factors are set via the [XYZ]COORD_CONV keywords of your graphics objects. The area of my plot fills the normalized space of [0, 0, 1, 1]. I'll add some buffer space around this viewplane to allow for annotations and the like.

I do remember seeing a specific example in the IDL documentation referring to coordinate transformations with object graphics. Try and find that section -- it really is buried in there somewhere.

> (4) I have followed David Fanning's advice on how to create symbols,
> but I still don't see any, just a line plot.
> Thanks in advance,

I believe this problem will be resolved when you take care of you viewplane issues. In your current situation, I sure wouldn't expect to see normal looking symbols. So, get the VIEWPLANE right and then we'll revisit this if need be.

-Mike
