Subject: Re: Setting data coordinates
Posted by David Fanning on Tue, 29 Jul 2003 23:05:09 GMT
View Forum Message <> Reply to Message

Marshall Perrin writes:

> I must confess to not fully understanding the "behind-the-scenes"
> details of how IDL handles converting between device, data, and normal
> coordinates in object graphics. I'd like to be able to explicitly
> set the conversion to data coordinates myself sometimes, and I'm not
> clear on how to do this. It's probably some system variable I need to
> set, I assume, but the docs have been less than forthcoming with which one...

No, there is no "system variable". You are thinking
direct graphics. :-)

The secret to understanding coordinate systems in object
graphics is to realize that there *are* no coordinate
systems in object graphics, except the one you make up
yourself, in whatever arbitrary units suits your purpose.
This arbitrary "view volume" is determined by the viewplane
rectangle and the near and far view planes.

Once you have decided on your coordinate system, you
just have to scale and translate all your objects into
that system. This is usually done by getting the natural
data range of the object (an image has a natural range
corresponding to the number of pixels in the image), then
doing the scaling and translating (I always use my
NORMALIZE routine because I've never been able to reliably
do all the math in this part), and, finally, putting it all
together with the [XYZ]Coord_Conv keywords.

> The application I have in mind is to display an image on screen, then
> set the data-to-device coordinate conversion properly so that I can
> then overplot points on my image (providing the points in data coordinates)
> and have them appear in the correct location on screen (in device coordinates).
> Should be simple, right? Any tips greatly appreciated.

So, this is the way I would do this. (Using my NORMALIZE routine,
which you can find on my web page.) Suppose you want to display
your image in arbitrary lat/lon coordinates of -180 to 180 and
-90 to 90.

```
 theView=Obj_New('IDLgrView', Viewplane_Rect=[-180, -90, 360, 180])
 theImage = Obj_New('IDLgrImage', myimage)
 theImage -> GetProperty, XRange=xr, YRange=yr
 xs = Normalize(xr, Position=[-180, 180])
```

```
  ys = Normalize(yr, Position=[-90,90])
  theImage -> SetProperty, XCoord_Conv=xs, YCoord_Conv=ys
```

Now you are ready to add whatever you like on top of the image.
Just specify the things you add in lat/lon coordinates.

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: Setting data coordinates
Posted by mperrin+news on Tue, 29 Jul 2003 23:09:05 GMT
View Forum Message <> Reply to Message

David Fanning <david@dfanning.com> wrote:
> Marshall Perrin writes:
>
>> I must confess to not fully understanding the "behind-the-scenes"
>> details of how IDL handles converting between device, data, and normal
>> coordinates in object graphics. I'd like to be able to explicitly
>> set the conversion to data coordinates myself sometimes, and I'm not
>> clear on how to do this. It's probably some system variable I need to
>> set, I assume, but the docs have been less than forthcoming with which one...
>
> No, there is no "system variable". You are thinking
> direct graphics. :-)

Wow, my brain clearly slipped a cog there. I *meant* to write
"direct graphics" after all.

That said, I think I'll try the object graphics approach you've so
thoughtfully described below. There's always more than one way
to skin a cat. :-)

 - Marshall

Subject: Re: Setting data coordinates
Posted by David Fanning on Tue, 29 Jul 2003 23:46:19 GMT

Marshall Perrin writes:

> Wow, my brain clearly slipped a cog there. I *meant* to write
> "direct graphics" after all.

Ah, right. :-)

Well, in the Catalyst Object Library Dave and I continue
to work on (without making any progress toward completion,
apparently) a data object always contains "pockets" (I
guess you could call them) for holding a coordinate
object and a color object. Before the data "draws itself"
in direct graphics, it "draws" its color object to set
up the proper drawing colors and it "draws" its coordinate
object to set up the proper coordinate system.

Draw widgets in our system know this about objects they
draw, of course, so they can always say to the data object
"given this point in the draw widget, tell me what the
corresponding point is in the coordinate system associated
with you". In this way, you can associate a map coordinate
object with an image object and always immediately find
out the lat/lon location of a click inside the image (or
anywhere else in the window).

Similarly, once a data object sets up a coordinate system,
any secondary object contained in the object's container
can be drawn in the same coordinate system. It gives a
sort of object graphics kind of feel to direct graphics
objects, without having to get terribly bogged down in
learning a completely new graphics system. Something to
keep in mind as IDL 6.0 comes out. :-)

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