Subject: Re: Scaling atoms & axes in object graphics
Posted by promashkin on Wed, 16 Aug 2000 07:00:00 GMT
View Forum Message <> Reply to Message

I must be really dumb but I don't understand how approach B is going to work. If axes ranges and data plots ranges are not normalized to screen coordinate scale, how are they going to show up on the normal scale with the axes? All my trial-and-error experience says it is not going to work. Or at least I'll say I don't know how to make it work without normalizing :-(
Anyway, I meant to say a track of plot components will still be needed, IMHO, for zooming (so that all of them are expanded or contracted) or for adding and deleting components while preserving the scale, so that the one with the largest range fits into the window. For that, I used two Container objects - one for Plots, another for Axes. Then, its easy to scale the needed components.
Maybe I am not fully plugged in on this, but I see but one general way of doing universal plot displaying (simple, not fly-through or something fancy).
Cheers,
Pavel

Mark Hadfield wrote:
>
> In the last few days I have been reconsidering my approach to building up
> scientific graphs in object graphics. By "scientific graphs" I mean a wide
> class of graphs in which data are represented geometrically in association
> with axes in 1, 2 or 3 dimensions. (This doesn't rule out the possibility
> that some aspects of the data are represented non-geometrically, e.g. by
> colour.) I am weighing the pros and cons of two different ways of handling
> scaling. Perhaps newsgroup readers would like to comment.

Snip - snip

Subject: Re: Scaling atoms & axes in object graphics
Posted by davidf on Wed, 16 Aug 2000 07:00:00 GMT
View Forum Message <> Reply to Message

Mark Hadfield (m.hadfield@niwa.cri.nz) writes:

> (*) Question for IDL expert programmers, why do I say that LOCATION
> represents a position perpendicular to the axis, when it has 3 components?

Isn't this the weirdest thing!? I think it took me about 10 hours to figure it out for the first time. The documentation is abysmal with respect to this keyword. I nearly gave up object graphics for good over this one keyword. I was resigned to changing values randomly to see (if I *could* see, one of

the *other* problems with object graphics) what effect it had.

> Thanks to Randall Frank for setting off this train of thought.

There is no question Randy knows object graphics. But I have to tell you, he and I are almost always on two different pages. He wants me to give up my Normalize function for good. I wish I could. But when I do I don't have a clue how to see graphics in my display window. :-(

Let's just say having a computer graphics class in my background wouldn't have hurt.

I'm really afraid to write too much of anything about object graphics because I know I do things in this diddly-shit way. But for the life of me I can't come up with anything better that I can understand. Probably says more about me than about object graphics, but there you go. :-(

Cheers,

David

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

Subject: Re: Scaling atoms & axes in object graphics
Posted by Paul van Delst on Wed, 16 Aug 2000 07:00:00 GMT
View Forum Message <> Reply to Message

Mark Hadfield wrote:
>
> The sole advantage I have cited for approach B is a biggy.
>
> What do others think? Is there an approach C I haven't thought of?

The only comment I have is to reiterate something you mentioned - the ability to scale axes independently is *very* important, e.g. zoom the x-axis but leave the y-axis alone. I do that as much as rescaling both (in DG).

The synopsis of your thinking was fun to read. Makes me want to

(re)start playing with OG.

paulv

--
Paul van Delst Ph: (301) 763-8000 x7274
CIMSS @ NOAA/NCEP Fax: (301) 763-8545
Rm.202, 5200 Auth Rd. Email: pvandelst@ncep.noaa.gov
Camp Springs MD 20746

Subject: Re: Scaling atoms & axes in object graphics
Posted by davidf on Wed, 16 Aug 2000 07:00:00 GMT
View Forum Message <> Reply to Message

Martin Schultz (martin.schultz@dkrz.de) writes (for example):

> Within a graph (level 3), everything should be on a fix scale, i,e.
> using something similar to
> Conv_Coord. However, "normal" coordinates for a graph would not be IDL's
> normal coordinates, but
> 0. and 1. would always correspond to the graph edges. The same holds for
> the panel, so that in order
> to position a graph on a panel you use the panel's "normal" coordinates.
> And once more for the
> page, only that here you may want to take care of the printable area or
> user defined page margins,
> so that 0. to 1. refers to the portion of the page that will receive
> some ink or toner when
> printed. For a screen window, on the other hand, you ignore the page
> margins, of course.

Martin, may I make a suggestion? Could you either
turn the word-wrap in your news reader off or
find that CR key more often? You are breaking up
so badly that it is distracting me from your
awfully good advice. I fear I'm not alone. :-)

Thanks,

David

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

Subject: Re: Scaling atoms & axes in object graphics
Posted by Martin Schultz on Wed, 16 Aug 2000 07:00:00 GMT
View Forum Message <> Reply to Message

Mark,

IMHO, you should stick to approach A for the reasons that you indicate. I had
once started to lay out a similar class structure and came to the following hierarchy:

0. (top) plot_collection ("album")
1. plot page (1 page equivalent to one window or one page of printed output)
2. panel (1 graphical entity that shows related things)
3. graph (1 plot or image which may have several curves or axes, etc)
4. molecules ;-) (axes, curves, contours, vector fields - each related to one data variable)
5. atoms (plot symbols, text labels, etc.)

Within a graph (level 3), everything should be on a fix scale, i,e. using something similar to
Conv_Coord. However, "normal" coordinates for a graph would not be IDL's normal coordinates, but
0. and 1. would always correspond to the graph edges. The same holds for the panel, so that in order
to position a graph on a panel you use the panel's "normal" coordinates. And once more for the
page, only that here you may want to take care of the printable area or user defined page margins,
so that 0. to 1. refers to the portion of the page that will receive some ink or toner when
printed. For a screen window, on the other hand, you ignore the page margins, of course.

Now, since these are many levels in a hierarchy, you probably want to be able to skip some of them,
similar to the IDLgr hierarchy of models and scenes. The minimum you would need to display data is
a graph, and if displayed or printed, it would fill the window or page. Then, to combine several
graphs in one figure for a publication (the typical (a), (b), (c), ...), you put the graphs into
a panel, and add some labels and perhaps even a figure caption. Now you want to produce a poster
or an overhead with two "figures" (or more, of course), then you combine several panels onto a

page. If each panel would only contain one graph, you could also store the graphs directly on
the page - all that matters is that both object types have a show method (and add, remove, ...).
At the very last, you can combine several pages in a "photo-album" to produce for example a
postscript file.

Then, if you really want to get fancy, you can play games: while a graph will generally be parallel
to the panel edges, you could freely rotate panels on the page (perhaps a solution to the postscript
upside down problem discussed a few days ago?). And by changing the panel or page size you have an
easy way of scaling the whole thing.

Doesn't this sound nice? The only drawback is that I will probably never have time to implement it.
I had started and got into details which led eventually to "variable" objects which are still at a
rather young development stage, but I am already using them for "file" objects which are still at a
rather young development, but I am using them for a "model evaluation" object which is still ...

But I promise to take a look at your objects before I start doing anything in this respect myself
(will be rather soon).

Cheers,
Martin

Mark Hadfield wrote:
>
> In the last few days I have been reconsidering my approach to building up
> scientific graphs in object graphics. By "scientific graphs" I mean a wide
> class of graphs in which data are represented geometrically in association
> with axes in 1, 2 or 3 dimensions. (This doesn't rule out the possibility
> that some aspects of the data are represented non-geometrically, e.g. by
> colour.) I am weighing the pros and cons of two different ways of handling
> scaling. Perhaps newsgroup readers would like to comment.
>
> Approach A: Use COORD_CONV
>
>

>
>
> Approach B: IDLgrModel::Scale & Translate, aka "COORD_CONV is evil"
>
>
>
> There is a major advantage to approach B:
>
> Once a set of atoms and axes has been put into a model, any further changes
> to the scaling of that model will not disturb the geometric relationship
> between the axes and atoms. With approach A, if you want to fiddle with the
> scaling of a graph after it has been set up, you have to EITHER: a) keep a
> list of all the atoms & axes at the top level of the graphics application
> and apply changes to the COORD_CONV properties of each one; OR b)
> have each axis keep track of all the atoms that are scaled to it and pass on
> any changes--then the graphics application just needs to keep track of the
> axes. I have been experimenting with the latter method via a class I call
> MGHgrMSaxis ("MS" = master-slave) and it's not as complicated as it sounds,
> but it's still a level of complication I'd like to avoid.
>
>
>
> Summary:
>
> The sole advantage I have cited for approach B is a biggy.
>
> What do others think? Is there an approach C I haven't thought of?
>
> Thanks to Randall Frank for setting off this train of thought.
>
> ---
> Mark Hadfield
> m.hadfield@niwa.cri.nz http://katipo.niwa.cri.nz/~hadfield/
> National Institute for Water and Atmospheric Research
> PO Box 14-901, Wellington, New Zealand

--
[[[[[[[[[
[[Dr. Martin Schultz   Max-Planck-Institut fuer Meteorologie   [[
[[Bundesstr. 55, 20146 Hamburg   [[
[[phone: +49 40 41173-308   [[
[[fax: +49 40 41173-298   [[
[[martin.schultz@dkrz.de   [[
[[[[[[[[[

Subject: Re: Scaling atoms & axes in object graphics

Posted by Mark Hadfield on Thu, 17 Aug 2000 02:28:52 GMT
View Forum Message <> Reply to Message

"Paul van Delst" <pvandelst@ncep.noaa.gov> wrote in message
news:399A963E.DBD142FD@ncep.noaa.gov...
> Mark Hadfield wrote:
>>
> The only comment I have is to reiterate something you mentioned - the
> ability to scale axes independently is *very* important, e.g. zoom the
> x-axis but leave the y-axis alone. I do that as much as rescaling both
> (in DG).

Yes, having thought about it overnight I think that is a *major* plus of approach A. If all the axes and atoms share the same data space, then every time you rescale in (say) the Y direction, all the X axes vanish off the top and bottom of the view (or congregate in the middle) and you have to reposition them.

Mark Hadfield
m.hadfield@niwa.cri.nz http://katipo.niwa.cri.nz/~hadfield/
National Institute for Water and Atmospheric Research
PO Box 14-901, Wellington, New Zealand