Subject: Re: Object graphic 3d Scatterplot
Posted by Ben Tupper on Wed, 22 Mar 2000 08:00:00 GMT
View Forum Message <> Reply to Message

Martin Schultz wrote:

> just an unsolicited question: wouldn't this be a good example for a new
> subclass of IDLgrPlot? You could extend the object definition by a new
> property (a pointer to an array DataClass for example) modify the
> SetProperty and GetProperty methods to allow changing/retrieving the
> DataClass values and change the Show (or whatever) method to use the
> DataClass values as a color coding for example. In pseudo code this
> would be something like:
>
> ClassgrPlot::Overplot,Data
>
>   ; get all DataClass values and loop using them as colors
>   uclass = uclass(uniq(*self.DataClass,sort(self.DataClass)))
>   for i=0L,n_elements(uclass)-1 do begin
>   ; define a suitable color or symbol/property object
>   color = i
>   w = where(self.DataClass eq uclass[i])
>   inherited->Overplot,self.Data[w],color=color
>   endfor
> end
>
>

Hello,

It sounds like a good idea... but waaay over my head.   I wonder if Mark Hadfield's MGHgrGraph already handles kind of a thing?

Ben

--
Ben Tupper

Bigelow Laboratory for Ocean Science
tupper@seadas.bigelow.org

pemaquidriver@tidewater.net

Subject: Re: Object graphic 3d Scatterplot
Posted by davidf on Wed, 22 Mar 2000 08:00:00 GMT

View Forum Message <> Reply to Message

I wrote few minutes ago:

> I'm quite sure this is something stupid on my part, but
> it always seems to happen to me when I am really SUPPOSE
> to be doing something entirely different.

Well, it was something stupid, to be sure. :-(

Anyway, here is my modified Simple_Surface program,
lightly modified to show a 3D scatterplot in object
graphics:

ftp://ftp.dfanning.com/pub/dfanning/outgoing/idl_course/scat ter_surface.pro

I started out with 32 points, bumped it up to 320 points without
difficulty. Then tried 3200 points. This slowed down quite
a bit, but I don't think the performance is totally unsatisfactory
even with this number of points. (And with this number of points,
the data is probably better viewed some other way anyhow, since
this is WAY too many points to visualize individually.) The
program as written shows 150 points and works great.

```
IDL> Scatter_Surface
```

Cheers,

David

P.S. Let's just say I'm reading Martin's latest post
in a new light now. :-)

--
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: Object graphic 3d Scatterplot
Posted by davidf on Wed, 22 Mar 2000 08:00:00 GMT

View Forum Message <> Reply to Message

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

> Please note that this is not a working example and that I have only gone

> 2 mm into object land so far. I'm just speculating ...

Uh, Martin, could you please speculate a bit on COLOR,
for goodness sake. That's where we seem to be stuck at the
moment. We'll worry about object refinements when the damn thing
works. :-(

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: Object graphic 3d Scatterplot
Posted by davidf on Wed, 22 Mar 2000 08:00:00 GMT

View Forum Message <> Reply to Message

Ben Tupper (tupper@seadas.bigelow.org) writes:

> Well, I found that I couldn't manage more than 10-100 points without
> seeing performance drop.

Well, in my little test program here I've got great performance
with something in the middle of this range. But at the moment
I can have the symbols drawn in any color I like, as long
as that color is black.

I'm quite sure this is something stupid on my part, but
it always seems to happen to me when I am really SUPPOSE
to be doing something entirely different.

If I figure it out, I'll let you know. In the meantime,
if anyone is interested in a neat, rotatable 3D scatterplot
with black symbols, contact me. :-(

Cheers,

David

P.S. Did you know that the new IDL 5.3.1 update for
Windows machines can improve the speed performance
by up to 40%? Could be why my programs seem to be

flying today. :-)

--
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: Object graphic 3d Scatterplot
Posted by Martin Schultz on Wed, 22 Mar 2000 08:00:00 GMT

View Forum Message <> Reply to Message

Ben Tupper wrote:
> Hello,
>
> Well, I found that I couldn't manage more than 10-100 points without
> seeing performance drop.   For each point, I defined a symbol obect and
> placed them all in an array. A lot depended on whether or not these
> points were draped across a surface with some kind of image texture.   A
> lot depends on the platform I suppose. After posting this, I'llprobably
> find out it is the programmer.   The data sets I am using are numbering
> 1000s of data points.   To be sure, a visualization with 5000-1000
> points can begin to obscure the different class distributions if the are
> more than just a few data classes (colors).
>
> It seems like such a simple thing (and wicked, as we DownEasters say,
> vital to datah vizulizin').   In direct graphics, I use colored (sized,
> etc.) data symbols all of the time to communicate some extra
> dimensionality to the data.
>
> Here's what all I know about the relationship between this attribute
> object (IDLgrSymbol) and its parent (IDLgrPlot, IDLgrPolyLine,...)
>
> So the plot object would have to perform umpteen GetProperty calls to
> umpteen symbol objects. (Which, by the way, the documentation doesn't
> make it very clear that this field can be set to an array of symbol
> objects.)

just an unsolicited question: wouldn't this be a good example for a new
subclass of IDLgrPlot? You could extend the object definition by a new
property (a pointer to an array DataClass for example) modify the
SetProperty and GetProperty methods to allow changing/retrieving the
DataClass values and change the Show (or whatever) method to use the
DataClass values as a color coding for example. In pseudo code this
would be something like:

```
ClassgrPlot::Overplot,Data

   ; get all DataClass values and loop using them as colors
   uclass = uclass(uniq(*self.DataClass,sort(self.DataClass)))
   for i=0L,n_elements(uclass)-1 do begin
      ; define a suitable color or symbol/property object
      color = i
      w = where(self.DataClass eq uclass)
      inherited->Overplot,self.Data[w],color=color
   endfor
end
```

Please note that this is not a working example and that I have only gone
2 mm into object land so far. I'm just speculating ...

Cheers,
Martin

```
--
 [[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[ [[[[[[[
[[ 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: Object graphic 3d Scatterplot
Posted by Ben Tupper on Wed, 22 Mar 2000 08:00:00 GMT

View Forum Message <> Reply to Message

David Fanning wrote:

> Ben Tupper (tupper@seadas.bigelow.org) writes:
>
>> That's ok for a small dataset; don't waste your time for
>> a large set.
>
> Humm. That's what I was thinking too. But I'm not clear
> by what I mean by "large". Can you share a bit more of
> your experience in this area, Ben?
>
>

Hello,

Well, I found that I couldn't manage more than 10-100 points without seeing performance drop.   For each point, I defined a symbol obect and placed them all in an array. A lot depended on whether or not these points were draped across a surface with some kind of image texture.   A lot depends on the platform I suppose. After posting this, I'llprobably find out it is the programmer.   The data sets I am using are numbering 1000s of data points.   To be sure, a visualization with 5000-1000 points can begin to obscure the different class distributions if the are more than just a few data classes (colors).

It seems like such a simple thing (and wicked, as we DownEasters say, vital to datah vizulizin').   In direct graphics, I use colored (sized, etc.) data symbols all of the time to communicate some extra dimensionality to the data.

Here's what all I know about the relationship between this attribute object (IDLgrSymbol) and its parent (IDLgrPlot, IDLgrPolyLine,...)

SYMBOL (Get, Set)
Set this keyword to a vector containing instances of the IDLgrSymbol object class. Each symbol in the vector will be drawn at the corresponding plotted point. If there are more points than elements in SYMBOL, the elements of the SYMBOL vector are cyclically repeated. By default, no symbols are drawn. To remove symbols from a plot, set the SYMBOL property equal to a null object reference.

So the plot object would have to perform umpteen GetProperty calls to umpteen symbol objects. (Which, by the way, the documentation doesn't make it very clear that this field can be set to an array of symbol objects.) This is, I suppose, the correct way to define the realtionship between attribute objects and parent objects as I read in Object-Oriented Design Heuristics by Arthur J. Riel (good advice on that David).   But, I seems to me that it might be equally right to make one GetProperty call to the attribute object for something like COLOR and then cycle through those colors.

Since I'm still an object newbie, maybe you could shed some light on this?

Ben

--
Ben Tupper

Bigelow Laboratory for Ocean Science
tupper@seadas.bigelow.org

pemaquidriver@tidewater.net

Subject: Re: Object graphic 3d Scatterplot
Posted by davidf on Wed, 22 Mar 2000 08:00:00 GMT
View Forum Message <> Reply to Message

Ben Tupper (tupper@seadas.bigelow.org) writes:

> That's ok for a small dataset; don't waste your time for
> a large set.

Humm. That's what I was thinking too. But I'm not clear
by what I mean by "large". Can you share a bit more of
your experience in this area, Ben?

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: Object graphic 3d Scatterplot
Posted by Ben Tupper on Wed, 22 Mar 2000 08:00:00 GMT
View Forum Message <> Reply to Message

Andrej Gapelyuk wrote:

> Hi all.
> I work with classification problem, and I would like to see how my data
> are distributed s in 3d space with possibilities of rotation. It should
> be possible to do with the object graphic. I?d like to see data for
> different classes like small particles of different colors . The closest
> tip which I have found was ? How to Create 3D Scatterplot? from David
> website, but unfortunately it is a direct graphic. Any suggestion.
>
>

Hello,

The object graphics path is the right one for 3d scatter plots that you
want to rotate. I have tried to perform the same kind of task using
datasets numbering in the thousands (of points.) Here's what I learned: be
advised that if you place you data into a IDLgrPlot object (using the
Zvalue and Use_Zvalue keywords) or IDLgrPolyLine, you can't make each point
a different color (to represent different classes) easily if you have a
large dataset. For each type of object, the symbol cattributes are defined
by another object, IDLgrSymbol. For each data point to have its own color,
you must define each point its own symbol object and then stuff them into
an object array. That's ok for a small dataset; don't waste your time for
a large set. I put in a request to allow the symbol object to accept an
arrays of attribute (like color, size, etc.) definitions rather than single
defintions. You might think about placing a request in the queue also.

Ben

--
Ben Tupper

Bigelow Laboratory for Ocean Science
tupper@seadas.bigelow.org

pemaquidriver@tidewater.net

Subject: Re: Object graphic 3d Scatterplot
Posted by davidf on Thu, 23 Mar 2000 08:00:00 GMT

View Forum Message <> Reply to Message

Struan Gray (struan.gray@sljus.lu.se) writes:

> I'm coming to this a bit late, but what the hell....

Yeah, but good information is *always* appreciated. :-)

> Polytype=1 (the default) gives you David's solution (though
> I've fiddled with the colour specification a bit - if you have a
> palette, why throw it away?).

Uh, mostly because you have done something idiotic and by
the time you figure out what it is, you have mucked the
code up something awful. :-(

I'm hoping to spend my long plane ride today reworking

these programs so I can offer them on my web page
in better shape. I appreciate these good suggestions,
Struan.

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: Object graphic 3d Scatterplot
Posted by Struan Gray on Thu, 23 Mar 2000 08:00:00 GMT
View Forum Message <> Reply to Message

I'm coming to this a bit late, but what the hell....

I played with this a bit when making a suite of routines to draw crystal structures using object graphics. My first approach was the polyline one that David has presented here. The final version simply creates a seperate graphics object for each data point.

The polyline approach is faster, both to create the dataset and to draw it on the screen. Using lots of different individual graphics objects is more versatile and makes some sorts of data picking and user interaction simpler.

Both approaches can be speeded up by limiting the number of symbols. In David's SCATTER_SURFACE each data point has its own symbol. If you can live with the reduced generality, use the fact that a palette for the colours limits the number of distinct symbols to 256. If you re-jig things so that the SYMBOL keyword refers to a 256-entry object array things speed up rather dramatically for 3000 objects. This code *should* drop nicely into David's example if you move the declaration of thisContainer above it:

```
   ; Create a color palette for coloring the symbols.

 thisPalette = Obj_New('IDLgrPalette')
 thisPalette->LoadCT, 5
 thisContainer->add, thisPalette
```

```
    ; Create the symbolarray

  npts = n_elements(z)
  nsymbols = min([256, npts])
  theseSymbols=ObjArr(nsymbols)
  FOR i=0,nsymbols-1 DO BEGIN
    if npts le nsymbols then colidx = zcolors[i] else colidx = i
    theseSymbols[i] = Obj_New('IDLgrSymbol', 4, $
      Color=colidx, Size=[0.05, 0.05, 0.05])
  ENDFOR
  if npts gt nsymbols then theseSymbols = theseSymbols[zcolors]
  thisContainer->add, theseSymbols

     ; Create Polyline object..

  thisPolyline = OBJ_NEW('IDLgrPolyline', x, y, z, $
    LineStyle=6, Symbol=theseSymbols, palette=thispalette)
```

If you are creating seperate graphics objects you can do the same thing by enclosing the plotted data point in an IDLgrModel and using the /ALIAS keyword to add the symbol objects which you create seperately. To illustrate this, and allow you to play with timings, I have crudely hacked David's program to accept two new keywords and placed it on my web server:

http://www.sljus.lu.se/stm/IDL/misc/scatter_surface.txt

The new keywords are NPTS (the number of data points you want) and POLYTYPE. Polytype=1 (the default) gives you David's solution (though I've fiddled with the colour specification a bit - if you have a palette, why throw it away?). Polytype=2 give you a graphics element for every data point. Polytype=3 minimises the number of graphics elements in 2. Polytype=4 minimises the symbols in David's approach, as above. 2 and 3 are slower than 1 and 4; 3 and 4 are faster than 1 and 2.

Finally, if you only have a few classes of data, and need only a few symbols, you can tidy up the programming by using a different polyline for each class (optionally, with SHARE_DATA). If you are joining up the symbols with lines, using the POLYLINES keyword to specify multiple sub-polylines is also useful. Both of these tricks make it easier to keep track of ownership of the data if you want to change it iteractively.

David Fanning, davidf@dfanning.com writes:

> Andrej Gapelyuk (gapelyuk@fvk-berlin.de) writes:
>
> The only disadvantage which I can see that all symbols are
> flat and non symmetrical in 3d, I would be prefer small
> spheres.
>
> As a matter of fact, it should be possible to use *ANY*
> kind of symbol you like, including 3D symbols (e.g. little
> shaded spheres), since the "value" of the DATA keyword for
> the symbol object can in fact be another model object.

This is true and works nicely. You can pass any IDLgrModel (or simple graphics object) to the IDLgrSymbol's INIT procedure and it will appear at each data point. Using a model lets you use stupidly complex symbols and, more usefully, lets you adjust their colours and other properties on the fly.

Struan

Subject: Re: Object graphic 3d Scatterplot
Posted by davidf on Thu, 23 Mar 2000 08:00:00 GMT

View Forum Message <> Reply to Message

Andrej Gapelyuk (gapelyuk@fvk-berlin.de) writes:

> Thanks a lot for everyone who help me to understand how to produce scatter plot
> in object graphic, especially for nice example from David. It works fine and
> fast for my data sets (several hundreds). The only disadvantage which I can see
> that all symbols are flat and non symmetrical in 3d, I would be prefer small

> already not so important.

As a matter of fact, it should be possible to use *ANY*
kind of symbol you like, including 3D symbols (e.g. little
shaded spheres), since the "value" of the DATA keyword for
the symbol object can in fact be another model object. (Or so I
am told by someone who should know.) The model can, of course,
contain any valid object hierarchy.

I may not have a chance to test this, since I'm off to
Europe today to teach an object programming course. But
you can bet that by the time I land on the other side of
the pond I'll have a new programming exercise for my
class. :-)

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: Object graphic 3d Scatterplot
Posted by Gapelyuk Andrej on Thu, 23 Mar 2000 08:00:00 GMT
View Forum Message <> Reply to Message

Thanks a lot for everyone who help me to understand how to produce scatter plot in object graphic, especially for nice example from David. It works fine and fast for my data sets (several hundreds). The only disadvantage which I can see that all symbols are flat and non symmetrical in 3d, I would be prefer small

already not so important.

Andrej Gapelyuk

Subject: Re: Object graphic 3d Scatterplot
Posted by Martin Schultz on Thu, 23 Mar 2000 08:00:00 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
>
> Martin Schultz (martin.schultz@dkrz.de) writes:
>
>> Please note that this is not a working example and that I have only gone
>> 2 mm into object land so far. I'm just speculating ...
>
> Uh, Martin, could you please speculate a bit on COLOR,
> for goodness sake. That's where we seem to be stuck at the
> moment. We'll worry about object refinements when the damn thing
> works. :-(
>
> Cheers,
>
> David

I'll leave color up to you, David: (1) you are much better and faster at
this, (2) the Hamburg sky is all grey for about 5 months now, so I don't
even know what colors are any more ;-)

Martin

--
[[[[[[[[[
[[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: Object graphic 3d Scatterplot
Posted by Mark Hadfield on Mon, 27 Mar 2000 08:00:00 GMT

View Forum Message <> Reply to Message

"Ben Tupper" <tupper@seadas.bigelow.org> wrote in message
news:38D90B6F.8F9377D7@seadas.bigelow.org...
> Martin Schultz wrote:
>
>> just an unsolicited question: wouldn't this be a good example for a new
>> subclass of IDLgrPlot? You could extend the object definition by a new
>> property (a pointer to an array DataClass for example) modify the
>> SetProperty and GetProperty methods to allow changing/retrieving the
>> DataClass values and change the Show (or whatever) method to use the
>> DataClass values as a color coding for example. In pseudo code this
>> would be something like:
>
> It sounds like a good idea... but way over my head.   I wonder if Mark
> Hadfield's MGHgrGraph already handles kind of a thing?

Some of the stuff I put in MGHgrGraph is way over my head too! But no, it
doesn't do anything like what Martin is suggesting. An MGHgrGraph is a form
of IDLgrView, and its methods are mostly concerned with fitting new graphics
atoms into a view: putting them in the right model, associating them with
axes, setting the scaling, applying fonts etc.

I've been following this thread with interest, but not very closely, and
don't have much to add, only to note that I have experimented with
subclassing graphics atoms' Draw methods and found that you can make an atom
look like anything you want. E.g. class MGHgrGLaxis

http://katipo.niwa.cri.nz/~hadfield/gust/software/idl/mghgrg laxis__define.pr
o

which hides an axis's labels on each Draw and substitutes its own.

It would be easy, for example, to create a symbol object that appeared in a different colour every time it was drawn. However the problem in applying this to the present problem is that this "smart symbol" would have to know which data point it is associated with on each Draw, and I don't think there would be any way of passing this information to it.

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