

---

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 object 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'll probably  
> 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 there are  
> more than just a few data classes (colors).  
>  
> It seems like such a simple thing (and wicked, as we DownEasters say,  
> vital to data visualization). 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(unique(*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      [[
[[

```

---