Subject: Re: Displaying 3-D vector fields
Posted by Struan Gray on Tue, 19 Nov 2002 10:21:43 GMT
View Forum Message <> Reply to Message

Rick Towler, rtowler@u.washington.edu writes:

> Ideally, as Mark has implemented in his barb plot, you'll have one model and
> one atom. Ignoring any parent models, IDL calculates a transform based on
> the one model and one atom transform (3x3 coord_conv transform) and applies
> it to all the vertices. Very efficient, but limiting.
>
> On the other end of the spectrum is a scene with 20k of my vector objects.
> Assuming we put our 20k models into a parent model, we need to do 40k matrix
> multiplies then apply each of those 20k transforms to the 4 verts that make
> up the vector. Not a model of efficiency but infinitely flexible.

We had a thread related to this a couple of years ago. If you search google.groups for "Object graphic 3d Scatterplot" (and/or my name) you will turn it up. There are a couple of other options.

The first is to use individual models for every object, but to use an alias. If your barb or arrow can be scaled without looking ugly (i.e. you don't mind the head scaling with the body) you can have a single barb object, and add it as an alias to 20k individual model objects. Each model object can have a different scaling, so your barbs can vary in size. This is actually a bit slower to plot than just creating a barb object for each point, but it is faster to create the model, and a lot faster if you want to animate it by, say, changing the barb colour for successive plots.

The second is to use the symbol keyword to a polyline plot. Even if you don't plot any lines, this turns out to be faster than creating multiple objects. I assume you strip out some overhead by having some of the loops through the object hiearchy done in native code.

The really cool trick though is to recognise that a 20k plot on any real reproduction medium is only going to have a limited subset of the 20k possible symbols. In the thread, the example was that if you use a palette to colour the symbols, there are only 256 possible colours, and so only 256 possible symbols, not 20k. In your case, I suspect you could select a small number of barb lengths and nobody would ever be able to tell.

This is of course, just a sneaky way of getting HISTOGRAM into the conversation: you create the desired characteristics for your barbs, and then use HISTOGRAM to distill them down into a subset. REVERSE_INDICES will tell you which data point uses which of the symbols in the subset. You can then use either the polyline+symbol technique, or make

individual models for each point and add the relevant symbol as an alias.

This may seem tedious, but for 20k objects every little helps.

Struan
