
Subject: Re: large 3D array plot
Posted by [Rick Towler](#) on Fri, 18 Aug 2006 18:25:25 GMT
[View Forum Message](#) <> [Reply to Message](#)

adisn123@yahoo.com wrote:
> I'm using linux/unix with remote X.

This is part of the issue. AFAIK, rendering while using remote X causes IDL to use the software renderer (but you'll want to verify this). That being said, even using the software renderer on my machine (3.6 GHz P4) I can view a cloud of 250k points reasonably well.

> And, when I said "(3, 210,000)", I meant 3 columns and 210,000 rows,
> thus 210,000 data points.
> These points plot a galactic image(stars, galaxies, clouds,...)

200k vertices aren't much for modern hardware. Part of the problem is that these points are haphazardly arranged which will slow rendering. I can render a million vertex surface faster than a 250k vertex random "cloud" since the verts that form the surface are arranged in an optimal fashion.

> I'm using symbols of spheres with linestyle = 6 thus, it does not
> connect each individual point, but rather show as points of spheres.

This compounds the issue immensely. Using spheres will add a large number of polygons to the scene thus slowing it down even further. If you followed the example for XPLOT3D, the default orb symbol has 402 vertices so now you're rendering ~85 million verts!

For this type of plot your approach (plotting a (hidden) line with symbols at the vertices) is probably the best way to go. Your real issue is the complexity of your symbol. Obviously you'll want to simplify it as much as possible. At the very least set the DENSITY keyword on your orb object to 0.2 which is the lowest you can set it to and still have a 3d object (it has 18 vertices). The simplest symbol you can get away with will be a tetrahedron (4 verts):

```
nverts = 210000  
seed = SYSTIME(/SECONDS)  
xyz = RANDOMU(seed, 3, nverts) * 1000
```

```
oTetra = OBJ_NEW('RHTgrPSolid', /TETRAHEDRON, $  
    RADIUS=0.2, COLOR=[75,125,230])  
oTetra -> GetProperty, OBJECT=pObj  
oSym = OBJ_NEW('IDLgrSymbol', pObj)
```

```
XPLOT3D, xyz[0,*], xyz[1,*], xyz[2,*], SYMBOL=oSym, $  
  LINESTYLE=6, /BLOCK
```

```
OBJ_DESTROY, [oTetra, oSym]
```

You can make your own tetrahedron or get RHTgrPSolid at:
http://www.acoustics.washington.edu/~towler/programs/rhtgrpsolid__define.pro

Another thing to consider would be to write your own viewer which HIDEs almost all of your data when you rotate the objects. You could do this by plotting the data in 2 IDLgrPolyline objects. One with some data, the other with lots. When a mousedown event is processed you hide the 'lots of data' polyline. At mouseup you unhide it. This probably would be an easy hack on XPLOT3D.

> I'm not sure what graphics card installed on my machine.

No need to check at this point, but if simplifying your symbol doesn't speed things up as much as you need, you may want to consider writing a program to plot your data that will work with the IDL VM and install that on your local machine. There still will be issues of configuring X to use hardware rendering which can be easy or hard but at least you'll have a better chance of viewing this data interactively.

-Rick
