
Subject: Re: large 3D array plot
Posted by [Rick Towler](#) on Fri, 18 Aug 2006 00:04:23 GMT
[View Forum Message](#) <> [Reply to Message](#)

What platform? Processor speed? If linux/unix are you using remote X?
And when you say "(3, 210,000)" do you mean you have 210,000 data points? What are you plotting (cloud, surface, head)? Are you using symbols? What do you set LINESTYLE to? Are you using hardware rendering? What graphics card is installed on your machine?

FWIW, your graphics card is probably not powerful enough to render your 200k vertices using xplot3d. And if you want to interactively move you rplot around, no other package is going to do much better.

Depending on what you are plotting, you may be able to eek out some more performance but without knowing what you are plotting it is hard to be specific.

-Rick

adisn123@yahoo.com wrote:

> Hi,
>
> My 3D vecotr(x, y, z)array is about (3, 210,000)size. Yeb. it's huge.
> So, when I plot in 3D using xplot3D, it takes a quite of time (more
> 10sec.), and it
> lags everytime I try to rotate.
>
> Is there anyone who knows better 3D plotting tool to display such a
> huge array in more efficient way?
>
> THanks.,
>

Subject: Re: large 3D array plot
Posted by [Kenneth P. Bowman](#) on Fri, 18 Aug 2006 00:38:40 GMT
[View Forum Message](#) <> [Reply to Message](#)

In article <1155854750.064537.283500@p79g2000cwp.googlegroups.com>,
adisn123@yahoo.com wrote:

> Hi,
>
> My 3D vecotr(x, y, z)array is about (3, 210,000)size. Yeb. it's huge.
> So, when I plot in 3D using xplot3D, it takes a quite of time (more
> 10sec.), and it

> lags everytime I try to rotate.
>
> Is there anyone who knows better 3D plotting tool to display such a
> huge array in more efficient way?
>
> Thanks.,

I tried this on my PowerBook G4, which is not a speed demon, and I can rotate the volume instantaneously.

```
xplot3d, randomn(seed, 200000), randomn(seed, 200000), $  
  randomn(seed, 200000)
```

Are you doing something different?

Ken Bowman

Subject: Re: large 3D array plot
Posted by [adisin123](#) on Fri, 18 Aug 2006 16:21:30 GMT
[View Forum Message](#) <> [Reply to Message](#)

I'm using linux/unix with remote X.
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,...)
I'm using symbols of spheres with linestyle = 6 thus, it does not
connect each
individual point, but rather show as points of spheres.
I'm not sure what graphics card installed on my machine.

How do I check those in my command line?

Rick Towler wrote:

> What platform? Processor speed? If linux/unix are you using remote X?
> And when you say "(3, 210,000)" do you mean you have 210,000 data
> points? What are you plotting (cloud, surface, head)? Are you using
> symbols? What do you set LINESTYLE to? Are you using hardware
> rendering? What graphics card is installed on your machine?
>
> FWIW, your graphics card is probably not powerful enough to render your
> 200k vertices using xplot3d. And if you want to interactively move you
> rplot around, no other package is going to do much better.
>
> Depending on what you are plotting, you may be able to eek out some more
> performance but without knowing what you are plotting it is hard to be
> specific.

>
> -Rick
>
>
> adisn123@yahoo.com wrote:
>> Hi,
>>
>> My 3D vecotr(x, y, z)array is about (3, 210,000)size. Yeb. it's huge.
>> So, when I plot in 3D using xplot3D, it takes a quite of time (more
>> 10sec.), and it
>> lags everytime I try to rotate.
>>
>> Is there anyone who knows better 3D plotting tool to display such a
>> huge array in more efficient way?
>>
>> THanks.,
>>

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 arrange 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

Subject: Re: large 3D array plot
Posted by [David Fanning](#) on Fri, 18 Aug 2006 18:43:06 GMT
[View Forum Message](#) <> [Reply to Message](#)

Rick Towler writes:

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

I think there might be some cut and paste issues in the descriptions of the keywords, but I got the idea. :-)

Cheers,

David

--

David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>
Sepore ma de ni thui. (Opata Indian saying, meaning "Perhaps thou speakest truth.")

Subject: Re: large 3D array plot
Posted by [Rick Towler](#) on Fri, 18 Aug 2006 20:33:53 GMT
[View Forum Message](#) <> [Reply to Message](#)

David Fanning wrote:

- > Rick Towler writes:
- >
- >> You can make your own tetrahedron or get RHTgrPSolid at:
- >> http://www.acoustics.washington.edu/~towler/programs/rhtgrpsolid__define.pro
- >
- > I think there might be some cut and paste issues in the
- > descriptions of the keywords, but I got the idea. :-)

Yeah, a little cut and paste problem... I noticed that this morning and fixed it locally but didn't upload the changes.

-r

Subject: Re: large 3D array plot
Posted by [adisin123](#) on Fri, 18 Aug 2006 22:53:06 GMT
[View Forum Message](#) <> [Reply to Message](#)

- > 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 familiar with IDLgrPolyline objects. but, it seems like it draws
a line between
points, which I don't want to do.

What do you mean by "HIDEs almost all of your data when you rotate the
objects."?

How come you want to hide part of your data while rotating in 3D, then

what's the point of viewing in 3D?

Maybe, I didn't get what you meant...

THanks for the great tip on the symbols. I've never thought that'd be
an issue.

Rick Towler wrote:

> 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 arrange 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
