
Subject: Re: An object graphics problem

Posted by [Karl Schultz](#) on Mon, 11 Aug 2003 19:50:14 GMT

[View Forum Message](#) <> [Reply to Message](#)

"Rick Towler" <rtowler@u.washington.edu> wrote in message
news:bh8n7n\$1doo\$1@nntp6.u.washington.edu...

>

> "Karl Schultz" wrote in message...

>>

>> "Gethyn Lewis" wrote in message...

>

>>> Now I want to cut the sphere with a plane that is perpendicular
>>> to the vector and goes through the spheres centre and then
>>> be able to look down the vector and see the data around the
>>> edge of the newly cut sphere. Any ideas?

>>

>> You may want to take a look at the CLIP_PLANES property.
>> It will be very easy to position a clipping plane at your
>> origin and orient it as you describe. You can use this
>> plane to clip the "front" half of the spheres and then you'll
>> see the concentric half-spheres nested behind the clip plane.
>> This might give you what you want if you want filled rings or
>> bands. If you want really narrow rings, then maybe a second
>> clip plane located a bit behind the front plane that clips
>> almost the entire back half of the spheres might work.

>

> If you are not going to be changing your clipping plane regularly you can
> also use MESH_CLIP() on your spheres in a manner similar to Karl's
> CLIP_PLANES method.

>

> It will be a bit of a toss up. The MESH_CLIP approach will take a bit
> longer to build the objects to display but will probably draw faster than
> the CLIP_PLANES method. If you are going to be changing your clipping
plane

> while you interact with the objects then the CLIP_PLANES method would be
> better since you wouldn't need to manage the vertex and connectivity data
at

> all.

More on the differences between the two approaches:

CLIP_PLANES clips the graphics as they are drawn to the display, while
MESH_CLIP gives you a new set of vertices and connectivity that you then
place in a graphic object. As Rick points out, you'll have to decide if you
want to do the extra work of creating clipped geometry and sticking it into
an object.

With MESH_CLIP, be sure to include your vertex colors with the AUXDATA

keywords. The clipping operation will generate new vertices and they'll need colors too. MESH_CLIP will interpolate the vertex colors on either side of the clipping plane to determine the colors of the new vertices. This will work fine for RGB vertex colors. If you are using indexed vertex colors, then this will work only as long as your palette is "smooth". If you have discrete colors such that a color index "somewhere in between" two color indices won't give you a reasonable color, then it won't work.

Using CLIP_PLANES will be easier to program, but may not draw as quickly, as Rick said. It will depend a lot on how many spheres you have and how fast your machine and graphics card are.

Karl
