Subject: Re: Tessellator
Posted by Karl Schultz on Mon, 12 Jul 2004 17:28:33 GMT
View Forum Message <> Reply to Message

"Ed Wright" <ed.wright@[null].jpl.nasa.gov> wrote in message
news:ccmdosfpc1@nntp1.jpl.nasa.gov...
>
> A question concerning IDLgrTessellator:
>
> I currently work with models of irregualar bodies (asteroids), these
> models consisting of a set of vertex cordinates and a plate-vertex
> assignment map. The IDL IDLgrPolygon facility greatly eased data set
> display and manipultation. While my data currently includes the
> plate-vertes map, I'd find it convenient to generate this map from the
> vertex coordinates.
>
> I attempted to create a map using only vertex data passed to the
> IDLgrTessellator object, but failed. The Tessellate method did produce a
> map, but the plates tended to pass through the body instead of a more
> nearest-neighbor configurations.
>
> I suspect I lack a complete understanding of IDLgrTessellator. Can
> anyone provide some pointers, or is this an inappropriate application of
> the tesselation function?

It doesn't sound like a good application of the tessellator to me.

The primary use of the tessellator is to convert a polygon that is planar and possibly concave or complex into a set of triangles that can then be rendered by IDLgrPolygon. One of the restrictions of IDLgrPolygon (as well as OpenGL and some other polygon renderers) is that the polygon must be planar and convex in order to render properly. These systems usually accomplish rendering a planar convex polygon by splitting the polygon into triangles anyway, but this is trivial if the polygon is planar and convex.

So, if you had a polygon description (set of vertices) that form a circle, you could just pass that to IDLgrPolygon and it would render properly.

But, if you moved one or more vertices towards the center to form a concavity, then you would have use the tessellator. You would give the tessellator the vertex list and get back a connectivity list that describes a set of triangles that cover the interior of the polygon using the same vertices. You would then store the original vertex list and the returned connectivity list in the IDLgrPolygon object and it will then draw correctly.

You can also give the tessellator a vertex list and a set of (planar) polygons described by your own vertex list. The tessellator will tessellate

each polygon into triangles and return a new connectivity list. If the polygons intersect or one is contained within another, the resulting intersections or inclusions can be considered as "holes" and the computed tessellation will reflect that (see the docs for more info). And finally if polygon countours intersect or self-intersect, new vertices may be formed, which you also have to take into account.

I can't tell what a plate-vertex map is from your posting, but I do know that passing a collection of vertices in 3D space to the tessellator isn't going to generate any useful output. It almost sounds like you are trying to reconstruct surfaces from the 3D vertex data, and that is a hard problem (see other recent postings). Perhaps QHULL and GRIDDATA may be useful if you are trying to do some sort of gridding.

Karl
