Subject: Re: DXF and Face3D
Posted by Thomas Launey on Tue, 05 Sep 2006 00:52:32 GMT
View Forum Message <> Reply to Message

Dear Karl,

I really needed to keep the Face3D entities because of some post-processing in a rendering software so I ended up building a DXF text output by reformating the vertice/connectivity arrays (i.e.: bypassing the Write method of the DXF object). Performance would probably be better by calling the object's method but entities would have to be converted to polygon anyway (?) so I abandonned this route.
Thank you very much for the clear explanations about the DXF object and about why the Tesselator was messing my mesh.
Regards,
Thomas

Karl Schultz wrote:
> If your original input data was in the form of an implicit quad mesh, then
> you might be able to write it to DXF as a quad mesh instead of a polygon
> with a connectivity list.
>
> MESH_SMOOTH just moves the vertices around and returns the modified vertex
> list without changing the connectivity. (I was a bit wrong about this
> in my last posting) So, you should be able to submit the new vertex list to
> DXF, since the shape of the mesh hasn't changed and you don't need a
> connectivity list. Use the POLYGON type and specify the MESH_DIMS instead
> of the connectivity list. Again, this only works if your original data can
> be specified as an implicit quad mesh.
>
> If you are dealing with general polygons with connectivity data, then you
> might just want to try it with the POLYGON enitity with connectivity
> list.
>
> My reading of the problem description is that IDL ran the polygon
> through the tesselator when it did not need to. This resulted in storing a
> polygon in the DXF file that had a different vertex list and connectivity
> list, as compared to the input. But, depending on the input, the mesh
> stored in the DXF file could be equivalent to the input mesh. If the
> input mesh had things in it like overlapping faces, self-intersections, or
> holes, the mesh stored in the DXF file might not be topologically
> equivalent to the input mesh.
>
> In other words, if your mesh is simple or regular, you might get away with
> it.
>
> Otherwise, you are probably out of luck if you do not upgrade to 6.2. The

> fix for 6.2 was pretty involved and I don't think that there are any other
> sneaky workarounds.
>
> Karl
>