Subject: Re: Compute_Mesh_Normals returns normal vector [0, 0, 0]
Posted by Rick Towler on Thu, 22 Jun 2006 17:26:18 GMT
View Forum Message <> Reply to Message

spidersapiens@gmail.com wrote:
> According to the online help, the function "Compute_Mesh_Normals"
> should return "a 3 x M array containing a unit normal for each vertex
> in the input array.". However when I used it in my program, it's
> returning a couple ZERO vector [0, 0, 0] though the rest are normal
> unit vectors. What does zero vector mean in this case or should I say
> this is an IDL bug?

Just a guess, but maybe a few of the verts are identical? I think for this to happen nVerts_per_poly-1 vertices would have to be the same. This might be in your data or it could be a subtle error in your connectivity array. Or it could be neither of these :)

-Rick

Subject: Re: Compute_Mesh_Normals returns normal vector [0, 0, 0]
Posted by James Kuyper on Thu, 22 Jun 2006 17:40:42 GMT
View Forum Message <> Reply to Message

spidersapiens@gmail.com wrote:
> According to the online help, the function "Compute_Mesh_Normals"
> should return "a 3 x M array containing a unit normal for each vertex
> in the input array.". However when I used it in my program, it's
> returning a couple ZERO vector [0, 0, 0] though the rest are normal
> unit vectors. What does zero vector mean in this case or should I say
> this is an IDL bug?

I'm not familiar with the algorithm used, though I assume it involves calculating cross-products of the edges which meet at the corresponding edge. However, you can imagine which situations might produce such a result without knowing the details of the algorithm.

If all of the edges are parallel to each other, all of the cross products would produce a 0 vector. If three edges meet at a vertex, and two of them are parallel to each other, a simple average would involve one cross product of exactly 0, and two cross products that exactly cancel each other. This is just the mathematical way of saying that the two pairs of edges give exactly conficting hints about the direction of the normal vector. The same can be said if four edges meet at a vertex, and any two of them are parallel.

Therefore, the thing you need to do is find out what the edges of the mesh are, that meet at the vertices where the normal vector comes out

0.