Subject: Re: point inside/outside of 3D object.
Posted by Karl[1] on Mon, 20 Jun 2011 19:57:25 GMT
View Forum Message <> Reply to Message

On Jun 20, 7:49 am, Wox <s...@nomail.com> wrote:
> On Fri, 17 Jun 2011 12:47:36 -0700 (PDT), Karl
>
> <karl.w.schu...@gmail.com> wrote:
>> I'm not sure you'd want to draw a tet with an ROI. A tet can be drawn
>> with a grPolygon. You would supply the 4 verts and then the
>> connectivity list which would be something like:
>
>> [3,0,1,2, 3,1,0,3, 3,2,1,3, 3,0,2,3]
>
>> The order is important to make all the faces facing "out". If any of
>> these are wrong, reverse the order. E.g., if the last tri is facing
>> the wrong way, change 3,0,2,3 to 3,3,2,0.
>
> Could you elaborate on that?
>
> I understand that the normal vector on each trianglular face of the
> polyhedron should point outwards. So if you take the three vertices of
> a triangle, they should be ordered so that when using the "right-hand
> rule", the normal points outwards.
>
> Lets mark the vertices of a tetrahedron Red, Green, Blue and Gray
> (http://tinypic.com/r/513fau/7). So the ordered vertices for each
> triangle should be
> back: Red-Green-Blue (equivalents: Green-Blue-Red, Blue-Red-Green)
> front-right: Red-Gray-Green
> front-left: Red-Blue-Gray
> bottom: Green-Blue-Gray
>
> So how do the quadruples come into play?

The first "3" is the number of indices that follow that contribute to
the next face. I believe that is how the connectivity list is defined
in IDLgrPolygon. Please see the docs for IDLgrPolygon.

If you put your vertices into an array v in this order: Red, Green,
Blue, Gray

Then your connectivity list that you'd pass to IDLgrPolygon along with
the vertex array v would be:

3,0,1,2, 3,0,3,1, 3,0,2,3, 3,1,2,3

Although I think you want Green-Gray-Blue for the bottom. You want

the normal for the bottom to point down, giving:

3,0,1,2, 3,0,3,1, 3,0,2,3, 3,1,3,2

which is equivalent to the first list I gave:

>> [3,0,1,2, 3,1,0,3, 3,2,1,3, 3,0,2,3]

taking different starting positions into account.
