Subject: Re: 3D array from surfaces?
Posted by Karl[1] on Wed, 14 May 2008 02:53:09 GMT
View Forum Message <> Reply to Message

On May 13, 5:52 pm, Rick Towler <rick.tow...@nomail.noaa.gov> wrote:
> Mike wrote:
>> On May 13, 1:00 pm, David Fanning wrote:
>
>>> Maybe you can get some inspiration from this article:
>
>>> http://www.dfanning.com/graphics_tips/mesh.html
>
>> Yep - That is just the opposite of what I want to do. What I'm
>> looking for is the inverse of Shade_Volume. I suppose this really
>> boils down to finding the intersection of a plane and a mesh
>> surface... I'll mull it over tonight and see what comes up...
>
> I have to admit that I am a bit confused as to what exactly you need but
> you'll probably want to look here:
>
> http://www.realtimerendering.com/int/
>
> My guess is that you'll want to sweep a plane orthogonal to the z axis
> from min(z) to max(z) testing for tri/plane intersections. Looking at
> that page, the code for tri/plane intersection testing isn't online but
> there are at least a few for tri/tri so you could simply use one of
> those and make the one tri really big.
>
> HTH's
>
> -Rick

It isn't super clear to me either, and the only way I can make sense
of it is if all the surfaces were closed, like spheres. Otherwise,
you could have points that you could not decide are inside or outside.

Assuming the surfaces are closed:

For each sample in the 3D volume, shoot a ray in any direction until
it hit something or leaves the volume. If it leaves the volume, mark
it as "outside". Determine the "side" of the triangle it hit
otherwise, by the relative direction of the ray and the triangle's
normal. (dot product? it's late) Then set inside or outside
accordingly.

Yes, and then running ISOSURFACE or ShadeVolume on the result should
give you back the surfaces you started with, or something close.