Subject: Re: Check if a point is inside a polygon mesh
Posted by Karl Schultz on Thu, 18 Aug 2005 16:10:07 GMT
View Forum Message <> Reply to Message

On Thu, 18 Aug 2005 03:53:19 -0700, photosalex@freenet.de wrote:

> Hello, All!
>
> Does anybody know a way to figure out if a 3D point lies inside a
> closed 3D triangular polygon mesh. The mesh is a tricky engineering
> construction of thousands of vertices with holes, stems, supports etc.,
> rather than a rectangular slab.
>
> The question arose basically from the task to find the coordinates of
> the intersection(s) of a ray with an arbitrary volumetric IDL model.
> I've searched the newsgroups but couldn't find a ready answer.
>
> If anyone provides me with information on either problems, I'd be very
> grateful.
>
> Cheers!

One way to do it is to create a point that you know is outside of the mesh. Use that point and the point in question to define a line segment. Then, perform a simple triangle/line-segment intersection check for each triangle in the mesh and keep track of the number of times the line intersects a triangle. If the total number of intersections is odd, the point in question is considered to be inside the closed mesh.

Karl

Subject: Re: Check if a point is inside a polygon mesh
Posted by cartik.sharma on Fri, 19 Aug 2005 01:32:28 GMT
View Forum Message <> Reply to Message

If the mesh is horribly non-convex, (and I believe it would be based on the description of the unstructured grid is..), it would be a good first step to calculate the convex hull of all points in the mesh using the quick convex hull procedure, QHULL and then checking if the point lies inside the convex hull using ray clipping. The qhull will also reduce the time taken to perform calculations.

-Cartik

photosalex@freenet.de wrote:
> Hello, All!
>

> Does anybody know a way to figure out if a 3D point lies inside a
> closed 3D triangular polygon mesh. The mesh is a tricky engineering
> construction of thousands of vertices with holes, stems, supports etc.,
> rather than a rectangular slab.
>
> The question arose basically from the task to find the coordinates of
> the intersection(s) of a ray with an arbitrary volumetric IDL model.
> I've searched the newsgroups but couldn't find a ready answer.
>
> If anyone provides me with information on either problems, I'd be very
> grateful.
>
> Cheers!

Subject: Re: Check if a point is inside a polygon mesh
Posted by Rick Towler on Mon, 22 Aug 2005 21:16:36 GMT
View Forum Message <> Reply to Message

Karl Schultz wrote:
> On Thu, 18 Aug 2005 03:53:19 -0700, wrote:

>> Does anybody know a way to figure out if a 3D point lies inside a
>> closed 3D triangular polygon mesh. The mesh is a tricky engineering
>> construction of thousands of vertices with holes, stems, supports etc.,
>> rather than a rectangular slab.
>>
>> The question arose basically from the task to find the coordinates of
>> the intersection(s) of a ray with an arbitrary volumetric IDL model.
>> I've searched the newsgroups but couldn't find a ready answer.
>>
>> If anyone provides me with information on either problems, I'd be very
>> grateful.

>
> One way to do it is to create a point that you know is outside of the
> mesh. Use that point and the point in question to define a line segment.
> Then, perform a simple triangle/line-segment intersection check for each
> triangle in the mesh and keep track of the number of times the line
> intersects a triangle. If the total number of intersections is odd, the
> point in question is considered to be inside the closed mesh.
>
> Karl
>

I'll add my 2 cents...

I have done a good bit of ray/tri intersection coding and can offer a

link and a suggestion. This page will be quite helpful:

http://www.realtimerendering.com/int/

It offers up links to a number of sources regarding different types of intersection. I have played around with Dave Eberly's intersection code as well as much of the Gems code. The Gems stuff is more academic, a starting point whereas Eberly's code is ready to go.

Lastly, you'll want to consider writing this code as a dlm for a couple of reasons. This type of problem requires a good bit of looping which can be a bottleneck in IDL. Also, much of the available code is written in C/C++ and in some cases "porting" would not be trivial.

-Rick
