
Subject: Re: IdlGrPolygon - Intersection with planes and lines
Posted by [Rick Towler](#) on Tue, 19 Nov 2002 23:54:06 GMT
[View Forum Message](#) <> [Reply to Message](#)

"Martin Downing" <martin.downing@ntlworld.com> wrote

> Hi all,
>
> Has anyone written code they wish to share which will give the
intersection
> of a polygon surface* with a 3D plane, and also the intersection of a
polygon
> surface with a line (3d ray)?
> One object graphics way to do the plane intersect could be to transform
the
> object to map the plane of intersection onto the $z = 0$, then use a very
> small zclip range and the result would be given in image form, however I
> would very much like the "exact" geometric solution, i.e the intersect as
a
> polyline for the plane and point(s) for the line (kind of graphics gems
type
> stuff)

Hi Martin,

Could you double mesh_clip the surface? The resultant mesh would still be a polygon though. Or, could you mesh_clip, where() the verts that lie on the clipping plane, and somehow reconstruct the polyline connectivity? That last bit might be sticky.

I've been playing with some polyhedron/box intersection code. Not what you are looking for, but in my search I have found a few resources which might be helpful if you can't find an all IDL solution.

There is the magic software archive:

<http://www.magic-software.com/Intersection3D.html>

which is a large C++ archive of many 3d graphics routines. I know he has a plane/ plane intersection test which returns the line of intersection which you could use if your surface was meshed with quads. I played around with some of the magic code in a dlm. You have to work a bit to get the data into a form that works with his classes and buy the book if you want any decent docs, there are few comments in the source.

Realtime Rendering:

<http://www.realtimerendering.com/#isect>

A section on intersection testing. Has some of the links I have provided.
Also links to this on the same site:

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

This web page has a table of static object intersection testing methods. It lists a few sources for triangle/triangle tests which you could use if your surface was meshed with tris.

There is also the Graphics Gems archive:

<http://www.acm.org/tog/GraphicsGems/>

This page might be helpful in tracking down which gem you'll need to get from your local library. I can tell you that there is little on this in Gems V. I also can tell you the Gems source code is where you'll probably want to start. It is implemented in C and tends to be very straightforward. Easier for us IDL'ers to chew on.

Hopefully this will get you started.

-Rick
