
Subject: Re: point inside/outside of 3D object.
Posted by [Wout De Nolf](#) on Fri, 17 Jun 2011 09:21:15 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Tue, 14 Jun 2011 12:01:25 -0700 (PDT), Junum <junshikum@gmail.com>
wrote:

> My questions are
> 1. I guess that a method defining a tetrahedron is wrong (i.e., px,
> py, pz).
> How can I define a 3D object consisting of several plane surfaces
> (e.g., cube) in IDLanROI?

Try the code below. "TetrahedronVertices" generates some polyhedron.
"test" checks whether a point falls within this polyhedron and plots
both.

As you can see, the point (green sphere) falls within the polyhedron
(red pyramid) although the IDLanROI says otherwise. I don't know
what's going on here ...

```
function TetrahedronVertices,r=r,phideg=phideg
; http://www.cs.umbc.edu/~squire/reference/polyhedra.shtml#tetrahedron
;
; r: any radius in which the polyhedron is inscribed
; phideg: goes from north pole 90 to -90 degrees

vertices=dblarr(3,4)

if n_elements(r) eq 0 then r=1d else r=double(r)
if n_elements(phideg) eq 0 then phideg=-19.471220333d else
phideg=double(phideg)

phi = !dpi*phideg/180.
theta120 = !dpi*120./180.
vertices[* ,0]=[0,0,r]
theta = 0.
for i=1,3 do begin
vertices[0,i]=r*cos(theta)*cos(phi)
vertices[1,i]=r*sin(theta)*cos(phi)
vertices[2,i]=r*sin(phi)
```

```

theta += theta120
endfor

return,vertices
end;function TetrahedronVertices

```

```

pro test

```

```

; Tetrahedron vertices
r=10
v=TetrahedronVertices(r=r)

```

```

; Close tetrahedron
v=v[*,[0,1,2,3,1,2,3]]

```

```

; Point
p=[0,0,0.]

```

```

; Inside/outside
object = Obj_New('IDLAnROI', v)
case object->containspoints(0.1,0.1,0.1) of
0: print,'Exterior'
1: print,'Interior'
2: print,'On Edge'
3: print,'On vertex'
endcase

```

```

; Plot object
oModel = OBJ_NEW('IDLgrModel')
oXAxis = OBJ_NEW('IDLgrAxis',$
0,title=OBJ_NEW('IDLgrText','X'),range=[min(v[0,*])-1,max(v[ 0,*])+1])
oYAxis = OBJ_NEW('IDLgrAxis',$
1,title=OBJ_NEW('IDLgrText','Y'),range=[min(v[1,*])-1,max(v[ 1,*])+1])
oZAxis = OBJ_NEW('IDLgrAxis',$
2,title=OBJ_NEW('IDLgrText','Z'),range=[min(v[2,*])-1,max(v[ 2,*])+1])
oModel -> Add, [oXAxis,oYAxis,oZAxis]
oModel -> Add, OBJ_NEW('orb',radius=r/10.,pos=p, COLOR=[0,255,0])
oModel -> Add, OBJ_NEW('IDLgrPolygon',v,
COLOR=[255,0,0],ALPHA_CHANNEL=0.5)
oModel -> ROTATE, [1, 0, 0], -90
oModel -> ROTATE, [0, 1, 0], 30
oModel -> ROTATE, [1, 0, 0], 30
XOBJVIEW,oModel

```

```

end;pro test

```

Subject: Re: point inside/outside of 3D object.
Posted by [Wout De Nolf](#) on Fri, 17 Jun 2011 11:43:42 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Fri, 17 Jun 2011 11:21:15 +0200, Wox <spam@nomail.com> wrote:

```
> ; Close tetrahedron  
> v=v[*,[0,1,2,3,1,2,3]]
```

Sorry, this part should be deleted. I was checking whether it could have something to do with "closing the tetrahedron" (rotate the object and notice that the bottom triangle is not filled).

Subject: Re: point inside/outside of 3D object.
Posted by [Junum](#) on Fri, 17 Jun 2011 15:45:35 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Jun 17, 6:43 am, Wox <s...@nomail.com> wrote:

```
> On Fri, 17 Jun 2011 11:21:15 +0200, Wox <s...@nomail.com> wrote:  
>> ; Close tetrahedron  
>> v=v[*,[0,1,2,3,1,2,3]]  
>  
> Sorry, this part should be deleted. I was checking whether it could  
> have something to do with "closing the tetrahedron" (rotate the object  
> and notice that the bottom triangle is not filled).
```

Thanks Wox.

It seems that IDLanROI works for 2D polygons only.

In case of 3D object, region of interest (i.e., object) is not defined properly.

I think points should be in a same plane.

Subject: Re: point inside/outside of 3D object.
Posted by [Karl\[1\]](#) on Fri, 17 Jun 2011 19:47:36 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Jun 17, 9:45 am, Junum <junshi...@gmail.com> wrote:

```
> On Jun 17, 6:43 am, Wox <s...@nomail.com> wrote:  
>  
>> On Fri, 17 Jun 2011 11:21:15 +0200, Wox <s...@nomail.com> wrote:  
>>> ; Close tetrahedron  
>>> v=v[*,[0,1,2,3,1,2,3]]  
>  
>> Sorry, this part should be deleted. I was checking whether it could  
>> have something to do with "closing the tetrahedron" (rotate the object  
>> and notice that the bottom triangle is not filled).
```

- >
- > Thanks Wox.
- > It seems that IDLanROI works for 2D polygons only.
- > In case of 3D object, region of interest (i.e., object) is not defined properly.
- > I think points should be in a same plane.

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.

One way to determine if a point is in the tet, or any closed object created out of triangles, would be to define a line between the point in question and any arbitrary point outside of the bounding box of the object. For each triangle in the object, determine if this line intersects the triangle. If the number of total intersections is odd, then the point is in the object. Note that this works for non-convex objects as well, as long as they are closed. You'll have to be careful about intersecting the object on a boundary between two or more triangles.

There may be a better way, but this is the basic brute-force approach.

If your object is just a tet, you can leverage that for a simpler solution. If the point is on the same inward-facing side of EVERY tri in the tet, then it is in the tet. Put another way, if the point is in the same inward-facing half-space of each triangle, it is inside. Think of the interior of the tet as the intersection of these half-spaces. I think that <http://steve.hollasch.net/cgindex/geometry/ptintet.html> describes the same idea mathematically.

Karl

Subject: Re: point inside/outside of 3D object.
Posted by [Junum](#) on Sat, 18 Jun 2011 18:34:42 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Jun 17, 2:47 pm, Karl <karl.w.schu...@gmail.com> wrote:
> On Jun 17, 9:45 am, Junum <junshi...@gmail.com> wrote:
>
>> On Jun 17, 6:43 am, Wox <s...@nomail.com> wrote:

```

>
>>> On Fri, 17 Jun 2011 11:21:15 +0200, Wox <s...@nomail.com> wrote:
>>>> ; Close tetrahedron
>>>> v=v*[0,1,2,3,1,2,3]]
>
>>> Sorry, this part should be deleted. I was checking whether it could
>>> have something to do with "closing the tetrahedron" (rotate the object
>>> and notice that the bottom triangle is not filled).
>
>> Thanks Wox.
>> It seems that IDLanROI works for 2D polygons only.
>> In case of 3D object, region of interest (i.e., object) is not defined
>> properly.
>> I think points should be in a same plane.
>
> 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.
>
> One way to determine if a point is in the tet, or any closed object
> created out of triangles, would be to define a line between the point
> in question and any arbitrary point outside of the bounding box of the
> object. For each triangle in the object, determine if this line
> intersects the triangle. If the number of total intersections is odd,
> then the point is in the object. Note that this works for non-convex
> objects as well, as long as they are closed. You'll have to be
> careful about intersecting the object on a boundary between two or
> more triangles.
>
> There may be a better way, but this is the basic brute-force approach.
>
> If your object is just a tet, you can leverage that for a simpler
> solution. If the point is on the same inward-facing side of EVERY tri
> in the tet, then it is in the tet. Put another way, if the point is
> in the same inward-facing half-space of each triangle, it is inside.
> Think of the interior of the tet as the intersection of these half-
> spaces. I think that http://steve.hollasch.net/cgindex/geometry/ptintet.html
> describes the same idea mathematically.
>
> Karl

```

Thanks Karl.

I wanted know whether IDLanROI::ContainsPoints can be applied to 3D case.

Subject: Re: point inside/outside of 3D object.
Posted by [Wout De Nolf](#) on Mon, 20 Jun 2011 13:49:09 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Fri, 17 Jun 2011 12:47:36 -0700 (PDT), Karl
<karl.w.schultz@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 triangular 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?
