Subject: Surface mesh
Posted by twhaw on Wed, 09 Apr 2008 11:26:43 GMT
View Forum Message <> Reply to Message

Good day.

I have created a 3D model consisting of points sampled on the outline
(surface) of a 3D object. The 3D object is initially translated so
that the origin is at its center of mass. The points are then sampled
in the spherical coordinates and converted back to the rectangular
coordinated and stored as a (3, n) array. The outline of the model is
sampled at a fixed interval longitudinally (every10 deg). Below is an
example of the sample points taken at 0 deg and 180 deg longitude so
that we can obtain the sample points around the object (similar to
cutting the object into halves through its center of mass).

```
              oooo
             o    o
             o     o
             o     o
   ooo       o     o
  o   o      o     o
 o     o  X  o     o
 o     o    o      o
 o      o  o       o
  o      oo       o
   o             o
     o          o
      o        o
        oooooo
```

where o represents the sample points and X represents the origin
(center of mass of the 3D object).

This process is repeated at 10 deg & -170 deg, 20 deg & -160 deg, 30
deg & -150 deg etc. Each longitude slice consists of the same number
of sample points (k). Hence I have a model consisting of 18 slices,
with each slice consisting of k sample points. These sample points are
stored as a (3, n) array, where n = k * 18 and each row corresponds to
one sample point (1st column gives its x-coord, 2nd column gives y-coord and 3rd column gives z-coord).

Now, I would like to create the surface mesh of the 3D model from the
sample points. I have used MESH_OBJ (type 0) but the resulting mesh
does not correspond to the surface of the 3D model. I have also
triangulated the points first to get the polygons and passed them to
IDLgrPolygon, but could not get the desired surface mesh.
Below are the codes that I have used:
; x, y and z are 1D vectors corresponds to the 1st, 2nd and 3rd column

of the (3, n) array defined above.

```
sph_coord = cv_coord(from_rect=transpose([[x], [y], [z]]), /
to_sphere, /degrees)
longiture = transpose(sph_coord[0, *])
latitute = transpose(sph_coord[1, *])
radius = transpose(sph_coord[2, *])
triangulate, longiture, latitute, tri, FVALUE=radius, /DEGREES,
SPHERE=myS
ntri = SIZE(tri,/DIMENSIONS)
ntri = ntri[1]
connect = LONARR(4,ntri)
connect[0,*] = 3
connect[1:3,*] = tri
oSurf = OBJ_NEW('IDLgrPolygon', TRANSPOSE(myS.XYZ), POLYGON=connect,
STYLE=2, $
SHADING=1, COLOR=[0,20,255])
oGroup = OBJ_NEW('IDLgrModel')
oGroup->ADD, oSurf
XOBJVIEW, oGroup, /block
```

These codes are adapted from one of the posts in this newsgroup and it works for spherical object. In fact, the resulting mesh resembles a sphere (which it shouldn't be).

If anyone has done this or something similar before, your help in this matter is highly appreciated.

Many thanks.