Subject: Re: texture map on cylindrical section
Posted by Rob[1] on Wed, 06 Apr 2005 00:24:05 GMT
View Forum Message <> Reply to Message

Rick,

Thanks a million. That works very nicely.

Rob

Rick Towler wrote:
>
> Yes, something as simple as this will work:
>
> zMax = MAX(verts[2,*], MIN = zMin)
> yMax = MAX(verts[1,*], MIN = yMin)
> texCoords = [(verts[2,*] - zMin) / (zMax - zMin), $
> (verts[1,*] - yMin) / (yMax - yMin)]
>
> You'll have to pick the plane you want to project your image onto.
In
> the case above I chose the zy plane. This may or may not be right
for
> you, I didn't really look.
>
> Now, maybe you already know how to do this and you realize that this
> won't apply the texture uniformly across the cylinder and that's why
you
> are posting...
>
> To do that I would create your cylinder orthoginal to the XZ plane
> centered about the origin (you can worry about transforming it
later).
> Then I would calculate the "horizontal" texcoords as a function of
the
> angle of rotation about the Y axis:
>
> yMax = MAX(verts[1,*], MIN = yMin)
> rotAngle = atan(verts[0,*],verts[2,*]) * !RADEG
> cMax = MAX(rotAngle, MIN = cMin)
> texCoords = [(rotAngle - cMin) / (cMax - cMin), $
> (verts[1,*] - yMin) / (yMax - yMin)]
>
> You'll probably have to dress that up a bit.
>
> So here it is all together. Enjoy!
>
> -Rick

```
>
>
>  pro cylindrical_sector
>  ; This procedure attempts to create a section of a cylindrical
>  ; shell.
>
>  ; Use extrusion to create the cylindrical section
>  ; First create an arc in the x-y plane
>  radius = 1.0 & height = 1.0
>  th_lo = 25.0 & th_hi = 180.-th_lo
>  n = 50 & dth = (th_hi-th_lo)/(n - 1.0)
>  th = th_lo+dth*findgen(n)
>  x = radius * cos(!dtor * th)
>  z = radius * sin(!dtor * th)
>  y = replicate(-0.5*height,n)
>
>  type = 5 ; extrusion
>  ; Create the array based on x and y
>  array = fltarr(3,n)
>  for i = 0,n-1 do array[0:2,i] = [x[i],y[i],z[i]]
>
>  ; Use MESH_OBJ to create the extruded section
>  mesh_obj, type, verts, polys, array, p1=1, p2=[0,1,0]
>
>  ; Create a model into which we'll stuff the POLYGON object
>  omodel = obj_new('IDLgrModel')
>
>  ; Create the image data to be texture-mapped onto the
>  ; POLYGON object.
>  ;image_data = bytscl(hanning(32,32))
>  image_data =
>   read_image(filepath('glowing_gas.jpg',subdir=['examples','da ta']))
>  help, image_data
>  oimage = obj_new('IDLgrImage',image_data)
>
>  ; Create the POLYGON object
>  osection = obj_new('IDLgrPolygon',verts,style = 2,polygons = polys, $
>      color = [255,255,255],texture_map = oimage)
>  omodel->add,osection
>
>  ;  easy method - not quite right
>  xMax = MAX(verts[0,*], MIN = xMin)
>  yMax = MAX(verts[1,*], MIN = yMin)
>  texCoords = [(verts[0,*] - xMin) / (xMax - xMin), $
>               (verts[1,*] - yMin) / (yMax - yMin)]
>  osection -> SetProperty, texture_coord=texCoords
>
>  xobjview,omodel, /block
```

```
>
>
>  ; better method
>  yMax = MAX(verts[1,*], MIN = yMin)
>  rotAngle = atan(verts[0,*],verts[2,*]) * !RADEG
>  cMax = MAX(rotAngle, MIN = cMin)
>  texCoords = [(rotAngle - cMin) / (cMax - cMin), $
>               (verts[1,*] - yMin) / (yMax - yMin)]
>
>  osection -> SetProperty, texture_coord=texCoords
>  xobjview,omodel, /block
>
>  obj_destroy, [omodel,oimage]
>
>  end
```
