Subject: Re: Cannot put an image back to the origin after rotating with T3D.
Posted by Karl Schultz on Fri, 29 Apr 2005 14:50:59 GMT
View Forum Message <> Reply to Message

On Thu, 28 Apr 2005 15:22:39 -0700, joe wrote:

```
>  Hello guys,
>
>
>  I created a torus with IDL but cannot put it back to the origin.
>  I don't really understand how t3d works with "rotation" and
>  "translation".
>  Everytime I rotate, the image seems to move from its original location.
>
>  Here is the code.
>
>  thanks in advance,
>  joe
>
>  pro torus_test, ri, ro, verts,polys, ndata=ndata
>
>
>  if ( not keyword_set(ndata) ) then ndata=80
>  nro=ndata & nri=nro/2   ; use ndata segments for bigger circle and
>                          ; use 1/2 number of segments for
>  smaller
>  circle.
>
>
>  theta = 2.0*!PI*findgen(nro)/nro
>  phi   = 2.0*!PI*findgen(nri)/nri
>
>
>  vx = fltarr(nro*nri)
>  vy = fltarr(nro*nri)
>  vz = fltarr(nro*nri)
>
>
>  polys = intarr(4,2*nro*nri)
>
>
>  np=0
>  for i=0,nro-1 do begin ; outer circle
>          i1 = (i + 1) mod nro    ;Polygon between segments (i,i1)
>          cx = cos(theta[i])*ro
>          cy = sin(theta[i])*ro
>          for j=0,nri-1 do begin ; inner circle
>                  j1 = (j+1) mod nri  ; Polygon between segments
```

```
>  (j,j1)
>                 rr = cos(phi[j])*ri  ; Radius of minor circle
>                 vx[i*nri+j] = rr*cos(theta[i]) + cx
>                 vy[i*nri+j] = rr*sin(theta[i]) + cy
>                 vz[i*nri+j] = ri*sin(phi[j])
>
>
>                 polys[0,np] =   [3, i*nri+j, i1*nri+j,
>   i1*nri+j1]
>                 polys[0,np+1] = [3, i*nri+j, i1*nri+j1,
>  i*nri+j1 ]
>                 np = np + 2
>           endfor
>       endfor
>  verts=[[reform(vx)],[reform(vy )],[reform(vz)]]
>  verts=transpose(verts)
>  return
>  end
>
>
>  t3d,/reset
>  verts=0
>  polys=0
>  torus_test,9,12,verts,polys
>  t3d,trans=[1.5,-0.5,-0.5]
>
>
>  t3d,ro=[5,-15,0]
>  verts=vert_t3d(verts)
>  print,!P.T
>  vmin = fltarr(3) & vmax = vmin  ;get min & max of each coordinate.
>  for i=0,2 do begin
>           v = verts[i,*]
>           vmin[i] = min(v)
>           vmax[i] = max(v)
>           endfor
>  !x.s = [-vmin[0],.9]/(vmax[0]-vmin[0] )/1.3 ;Set up data scaling to
>  normalized
>  ;                              coordswhich is the cube
>  [0,1],[0,1],[0,1]
>  !y.s = [-vmin[1],.9]/(vmax[1]-vmin[1] )/1.3
>  !z.s = [-vmin[2],.9]/(vmax[2]-vmin[2] )/2.8
>  surfr,ax=8              ;view from 45 degree orientation
>  erase
>  print, !P.T
>  ;  Monochrome?
>  if !d.n_colors le 2 then top = 255 else top = !d.n_colors -1 < 255
>  b = polyshade(verts,polys,/t3d,/da ta, xsize=512,ysize=512, top=top)
```

> tv,b
>
>
> end

Normally, if you want to rotate an object about its center, you would first translate the object so that its center is at the origin, rotate, then translate back to where it was.

In this case, the torus that you get back from torus_test is already centered at the origin, so all you have to do is apply the rotation.

Also, the surfr procedure resets the transform, so you should think about why you are using that.

If I remove the t3d call that applies a translation and the surfr call, I *think* I get what you might be expecting.

Karl
