
Subject: Re: Image plot on back wall

Posted by [raouldukey](#) on Tue, 23 Nov 1999 08:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

Ok...once again, I have to apologize for the poor form of my post. You can tell that I am new to posting on newsgroups. Anyway, I am behind a major firewall, so I can't put a nice ftp link to the file, hence my cut and paste technique to put the code in a usable form. However, the editor hacked it all up, so some lines that are meant to be commented out aren't. Let's try this again..... (hopefully I will soon get the hang of this as I think this is a useful forum).

```
-----  
; NAME:  
;   newsgroup  
;  
; PURPOSE:  
;   The purpose of this program is just a demonstration  
;   of what I did to put an image plot on the back wall  
;   of a cube. Basically, all I did was exchange the yz  
;   axis at the correct point in the show3.pro routine  
;   (sorry RSI). I will cut and paste RSI's routine in  
;   to this code, and mark my modications. Sorry about  
;   the lack of comments, but complain to RSI, as I can't  
;   figure out their code well enough to comment on it.  
;  
;  
; AUTHOR: Raouldukey  
;  
;  
; CALLING SEQUENCE:  
;   EXAMPLE_SURFACE, data  
;  
; REQUIRED INPUTS:  
;   None. Fake data will be used if no data is  
;   supplied in call.  
;   (Stole this from David - Thanks!)  
;  
; OPTIONAL INPUTS  
;  
;   data: A 2D array of surface data.  
;
```

```

pro newsgroup,image

;-----

; Need fake data?

IF N_Elements(image) EQ 0 THEN BEGIN
  image = BeselJ(Shift(Dist(40,40),20,20)/2,0)
ENDIF

;-----

set_plot,'z' ;Set graphics device to the Z buffer

;-----

;Get data dimensions

sizer = size(image)
numberx = sizer[1] ;columns
numbery = sizer[2] ;rows

if n_elements(x) eq 0 then x = findgen(numberx)
if n_elements(y) eq 0 then y = findgen(numbery)

;-----

img = image

xx = x
yy = y

;Tweak Values to get it to look
;the way I like (axis angles, max values,etc.)

ax = 40
az = 30
minz = min(img)
maxz = 3*max(img)

set_shading,values=[0,150],light=[0,0,1]

```

```
notick=[' ',' ',' ',' ',' ',' ',' ',' ']
```

```
;-----
```

```
;Ok....below here is where I start the copyright  
;infringement. The following all belongs to RSI,  
;and I have just made modifications to their routines.  
; I have stripped it down to the bare bones, just  
; to make it more obvious what I have done. You can  
; use this as an example to modify the full routine  
; of show3.pro
```

```
; Also, I have switched everything to shaded  
; surfaces because they  
; just look more cool for my data.
```

```
;-----
```

```
; Call shade_surf to get the 3D coordinate matrix
```

```
shade_surf,img,xx,yy,/save,xstyle=1,ystyle=1,$  
    zaxis=0,zrange=[minz,maxz],$  
    zstyle=1,az = az,ax=ax,ztickname=notick,$  
    /nodata
```

```
;x locations of corners  
xorig = [x[0],x[numberx-1],x[0],x[numberx-1]]  
;y locations of corners  
yorig = [y[0],y[0],y[numbery-1],y[numbery-1]]
```

```
; normalized x coordinate  
xcoor = xorig * !x.s[1] + !x.s[0]  
; normalized y coordinate  
ycoor = yorig * !y.s[1] + !y.s[0]
```

```
;-----
```

```
; I added the following line to rotate the  
; xy axis to the vertical as the show3 routine  
; projected it to the xy plane already. Obviously,  
; the proper way to do this would be to figure  
; out the coordinates of the back wall (xz plane)  
; and use polywarp to warp it there. I couldn't  
; work out how to do this correctly, so good luck!
```

```
t3d,/yzexch
```

```
;-----
```

```
;Back to the show3.pro routine with all of  
; its great comments  
; (thanks RSI)  
;  
; #!P.T is the transformation matrix we set up  
; with shade_surf routine and the xcoor,ycoor  
; correspond to the pixel coordinates of our surface
```

```
p = [[xcoor],[ycoor],[fltarr(4)],[replicate(1,4)]] # !P.T
```

```
; Scale U coor to device  
u = p[* ,0]/p[* ,3] * !d.x_vsize  
; and v  
v = p[* ,1]/p[* ,3] * !d.y_vsize
```

```
u0 = min(u)  
v0 = min(v) ;lower left corner
```

```
sizeu = max(u) - u0 + 1  
sizev = max(v) - v0 + 1 ;Size of new image
```

```
fact = 1  
miss = 0
```

```
;-----
```

```
; Figure out kx, ky for our desired warped surface
```

```
polywarp,xorig,yorig,(u-u0)/fact,(v-v0)/fact,1,kx,ky
```

```
warpedimage = poly_2d(bytescl(img),kx,ky,$  
keyword_set(interp),sizeu,sizev,missing=miss)
```

```
; -----
```

```
; We now have the image warped vertically. It doesn't  
; seem to be perfect, but not too bad. Now...slide  
; it to the back of the cube with the following  
; numbers in the tv command.
```

```
tv,warpedimage,63,190,xsize = sizeu,ysize=sizev,order=0
```

```
;-----
```

```
; Draw the shaded surface in front of our image
```

```
shade_surf,img,xx,yy,/save,xstyle=1,ystyle=1,$  
  zaxis=0,zrange=[minz,maxz],$  
  zstyle=1,az = az,ax=ax,/noerase,$  
  ztickname=notick
```

```
;-----
```

```
;Get the image from the Z-buffer  
; Adjust device for what you need  
; (PS, Xwin, windows...etc)
```

```
finalimage = tvrd()  
set_plot,'win'
```

```
;-----
```

```
;Draw the final image to screen
```

```
tv,finalimage
```

```
end
```

Sent via Deja.com <http://www.deja.com/>
Before you buy.
