Subject: Z-Buffer
Posted by J.D. Smith on Tue, 11 Feb 1997 08:00:00 GMT
View Forum Message <> Reply to Message

Perhaps I misphrased my question...

I don't mean to break up the contest, but my real question is as follows: Suppose you create an image in the Z-buffer, but put no axes on it. This image is read from the buffer into a variable. Since I had already discovered Mr. Savoie's solution to the character size problem (which is what is actually causing the misalignment of axes in Mr. Fanning's posited conundrum), set the character size appropriately. And now the fun part.... Switch to the postscript device, display the z-buf image in a position you'd like, and *now* put axes on top. The reasoning for this is obvious: unless you'd like to set your z-buffer resolution to that of your postscript device, the text of the axes will not survive that foul temptress the z-buffer with even close to satisfactory quality.

Here's some summarizing code (for those whose decompose the written word more readily into code fragments and blocks than sentences and paragraphs):

```
dev=!D
set_plot,'Z'
!P.BACKGROUND=!D.N_COLORS-1 ; to get white background
device,set_character_size=[dev.x_ch_size,dev.y_ch_size]
;; put 3 surfs atop eachother ... the z's and e's are concocted elsewhere
shade_surf,zmt,e1,e2,shades=bytscl(zmt,TOP=!d.table_size), $
 charsi=2.25,/save,az=20,zrange=[0,1.3],xst=4,yst=4,zst=4
shade_surf,zm1,e1,e2,shades=zm1*0B+!d.table_size/3,/T3D, $
 xst=4,yst=4,zst=4,charsi=2.25,/NOERASE,zrange=[0,1.3]
 shade_surf,zm2,e1,e2,shades=zmt*0B+!d.table_size/4,/NOERASE, /T3D, $
 xstyle=4,ystyle=4,zstyle=4,charsi=2.25,zrange=[0,1.3]
;; get the pixture out
a=tvrd()
device,BITS=8, xsize=8.5,ysize=11.,xoffset=0.,yoffset=0.,/inches
!P.MULTI=[0,1,3,0,0]
tv,a,.5,5,xsize=5,ysize=5,/inches
;; now add the axes....
shade_surf,zmt,e1,e2,/T3D,xtitle='e!d1',ytitle='e!d2', $
 ztitle='Normalized Merit',/NODATA,/NOERASE, $
 TITLE='Ritchey-Chretian Optimization',charsi=2.25,zrange=[0,1.3]
device,/close
set_plot,'X'
```

Of course, the axes fill the postscript page... here is the solution I attempted:

Pass the POSITION keyword to the final shade_surf call. This is what I think will eventually work, but the problem is, you must set x,y, *and* z positions, whereas the region you've just displayed is contrained in x and y (since the z-buffer image was "flattened" by tvrd() and tv() controls the offsets and sizes). Setting the x and y only (e.g.,POSITION=[.5/8.5,5./11,8./8.5,10./11.]) will only shrink the x and y axes. What I therefore need is an algorithm which, given a 2-D region of interest within the plotting window, computes the 3-D position needed to overlay that image with another 3-D plot, given the x,y,z margins, charsizes, etc. (and !P.T). No simple matter I fear.

The real issue here is the incongruence of the image and plotting parameters specifiying locations inside the plotting region. If these were more unified, perhaps one could say: put a 3-d plot in this region exactly as if it were the entire region (as in the examples presented previously in this thread).

Anyway, the solution is no longer critical to me... but just thought I'd set the record straight.

I think Matthew should receive the prize... but I suppose there are a few days left.

Thanks,

JD
