
Subject: Re: Mapping on Objekts...

Posted by [J.D. Smith](#) on Wed, 19 Mar 1997 08:00:00 GMT

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

David Fanning wrote:

>
> I've never really tried this with a zylinder, but I think I could get it to
> work. You do it in the C-buffer, with the Pattern and Image_Coord
> keywords. (Image_Coord matches the coordinates of the image in
> Pattern with the vertices specified in the positional argument
> to Polyfill. For a zylinder you may have to do the rendering in narrow strips
> to make the edges more or less a smooth curve. (Guess this depends on
> how your zylinder is rendered on the display.)
>
> Cheers!
>
> David
>
> P.S. You can find a good description of how to do these Z-buffer
> tricks in my soon-to-be announced IDL Programming Guide. I am
> very pleased with the new graphics sections. It may be the
> best material I have ever written on IDL techniques. :-)

Well, it's a little harder than you might think. My difficulty has to do with setting up the coordinates in polyfill so that the region you want is actually specified. For a cylinder, I broke it up into semi-rectangular sections (like the outer crust on slices of cheesecake). The problem is, I couldn't find a way to get polyfill to accept vertex coordinates to define the cylinder *sides* (not the circular top or bottom). I tried all sorts of variations, even full rectangular indices formed with:

```
;; form the indices to get [0,0,1,1,0,1,1,2,2,1,2,2,3,3,2...]
;flag=0
;for i=0,N-1 do begin
;  if flag eq 1 then begin
;    inds=[inds,[i,i,i+1,i+1,i]]
;  endif else begin
;    flag=1
;    inds=[i,i,i+1,i+1,i]
;  endelse
;endfor
```

None of these gave the desired results. The best I could do is every other section (like a circular of teeth with gaps). So, instead of trying to give polyfill the entire list at once, I filled each section, one at a time. This worked just fine. The code follows. Simply put it in a file and say .run file. Here are some caveats: If N (the number of sections) is larger than the number of columns in your data, the code will work, but will not make pretty results. You could congrid your array to be larger, if necessary (or reduce N). Also, note that, for some reason, passing polyfill the *subset*

of
the array and giving IMAGE_COORD's with respect to the subset is much faster than passing the
entire
array and using the real coordinates (but why I can't imagine). I got good results with $N \approx \text{data columns}/2$.

```
:: BEGIN CYLINDER MAP CODE
```

```
:: set up 3-D scaling  
scale3,xrange=[0,1],yrange=[0,1],zrange=[0,1],AZ=0  
N=100 ;number of sections of cylinder
```

```
:: get x and y coordinates of a circle -- you can change the circle's  
:: shape here -- a fun example: replace the yc=.5.. statement by the  
:: commented statement to its right  
xc=findgen(N)/(N-1)*PI*2 & yc=xc  
xc=.5*(1+cos(xc+!PI/2))  
yc=.5*(1+sin(yc+!PI/2)) ; yc=.5*(1+sin(yc+!PI/2)+.1*sin(N/10*yc))
```

```
:: get some example data  
file=filepath('people.dat',subdir='examples/data')  
openr,un,/get_lun,file  
d=bytarr(192,192)  
readu,un,d  
free_lun,un  
s=size(d)
```

```
:: go to Z-buffer  
oldname=!D.NAME & set_plot,'Z'  
erase
```

```
::draw each section individually  
for i=0,N-2 do begin  
  ;; setup coordinates defining the current section of the cylinder  
  x=[xc(i),xc(i+1),xc(i+1),xc(i)]  
  y=[yc(i),yc(i+1),yc(i+1),yc(i)]  
  z=[0,0,1,1]  
  ;; setup image coordinates to pin to vertices overlaying this section -- screwy for  $N > s(1)$   
  left=round(float(i)/N*(s(1)-1)) & right=round(float(i+1)/N*(s(1)-1))  
  top=s(2)-1 & bot=0  
  ;; fill the section with image piece.. IMAGE_INTERP makes things smooth  
  polyfill,x,y,z,/T3D,PATTERN=d(left:right,bot:top), /IMAGE_INTERP,$  
  IMAGE_COORD=[[0,0],[right-left,0],[right-left,top],[0,top]]  
endfor
```

```
im=tvrd() ; read in the image  
set_plot,oldname ;switch back to your display device
```

```
tv,im  
end
```

```
:: END CYLINDER MAP CODE
```

Hope this helps.

JD
