Subject: Re: 3D Surface Problem
Posted by btupper on Tue, 09 Apr 2002 01:04:54 GMT
View Forum Message <> Reply to Message

Hello,

I think Rick has the right idea about providing the xyz locational
info for each vertex. Since you are showing images projected on a
map, then the original x and y dimensions are lifted from a model of a
spheriod and, in their raw unprojected form, will produce a stretched
flat map. But all that is out of my league.

On Mon, 8 Apr 2002 10:42:27 -0600, Ken Mankoff
<mankoff@I.HATE.SPAM.cs.colorado.edu> wrote:

>> Please post some more detail or code.
>
> I think the problem is a few things
>
> 1) Walls are appearing at the edge of the DEM. Mars has a lot of its
> surface at below sea-level (negative altitude). But even if I say
> "dem = dem - min(dem)", so everything should be positive, I still
> get the "walls"
>
> 2) The images always come out square. I am pretty sure this is just a
> keyword I am missing, but I do not know which one and which object it
> belongs with (surface? window?).
>
> 3) The vertical scaleing is always "0 to 1" in the IDLgrModel (I
> think). This looks good with maps that cover a large vertical area
> (say, Olympus Mons or Valles Marineris). But if the map is of a
> relatively flat area (somewhere in the northern low-lands), and the
> DEM covers a few hundred meters, then those few hundred meters get
> streched vertically, and it appears warped.
>

I have an graphic object model that simplifies process of anisotropic
scaling of dimensions
(www.tidewater.net/~pemaquid/axesgroup__define.pro) Basically, it
automatically builds simple axes for you. You then take the scaling
parameters (*conv_coord, normailzed or not) and pass them along to the
surface object. It maybe helpful as an example. You will need Martin
Schultz's object code distribution and David's NORMALIZE function.
The relevant method is ScaleAxes in AXESGROUP. You can play with the
three element Isotropic property which specifies relative scaling
among the X, Y and Z dimensions.

Here's a mock up of how you might use it. Call both before destroying
the first so you can see how setting parts of the underlying dem to
NaN gets rid of that ugly edge.

```
IDL> vmars2,axes1
IDL> vmars2,axes2, /nan


;;;;START

PRO vmars2, axes, nan = nan

dem = fltarr(35,45)
dem[2:31,2:31] =  dem[2:31,2:31]-dist(30) - 8.0
img = bytscl(dem)

if keyword_set(nan) then begin
 a = where(dem GT -8.0, cnt)
 if cnt gt 0 then dem[a] = !Values.f_nan
EndIf


;;; image coordinate setup
iDims = size( img, /DIM )

;;; surface coordinate setup
s = (sz = SIZE( dem,/dim ))
maxz = MAX( dem, MIN=minz,/nan )

Axes = OBJ_NEW('axesgroup', [0, s[0]], [0, s[1]], [minZ, maxZ], $
 style = [15,15,15], $
 xcolor = [255,0,0], $
 ycolor = [0,255,0],$
 zcolor = [0,0,255], $
 exact = [1,1,1], $
 /normalize)

scale = axes->GetScale()
;;; image -> surface coordinates

image = obj_new( 'idlgrimage', img )
surface = obj_new( 'idlgrsurface', $
                   dem, $
                   style=2, $
                   color=[255,255,255], $
                   texture_map=image, $
                   shading=1, $
```

```
                    xcoord_conv=scale[*,0], $
                    ycoord_conv=scale[*,1], $
                    zcoord_conv=scale[*,2] )

Axes->Add, surface

xObjView, axes, background = [0,0,0]
end

;;;;END
```

> 4) White pixels are appearing in the images.

Are you sure those aren't reflections of grey hair in your monitor?

Ben