Subject: Re: 3D Surface Problem
Posted by Ken Mankoff on Mon, 08 Apr 2002 16:42:27 GMT
View Forum Message <> Reply to Message

> "Ken Mankoff" <mankoff@I.HATE.SPAM.cs.colorado.edu> wrote :
>> I am using IDL Object Graphics to combine a Digital Elevation Model
>> (DEM) with a photograph in an attempt to get a pseudo-realistic 3D
>> view of a surface.
>>
>> I have it mostly working, but have a shear effect that I cannot get
>> rid of... I do not fully understand this whole 3D viewpoint thingey.
>> Also, my images shapes vary. Sometimes they cover a 1x1 degree lat/lon
>> grid, other times its a 1x10 (it is not always two squares I am
>> putting together, although the DEM and the image are always the same
>> shape). Also, the DEM sometimes has a range of a few hundred meters,
>> and other times a few thousand meters...
>
> The image shape shouldn't matter as long as you have the correct texcoords
> and for the most part they look good. It looks like you might have a white
> background for your non-square images? Is that where the "noise" in the
> texture comes from or is that a rendering issue?
>
> Where do you think the problem is? Can you render the object correctly and
> then you botch the transform? Or is the problem before you render a view?
> There could be a number of possible issues with your examples.
>
>> Ideally, I would love a function to do this for me that is canned and
>> pre-written. (I will give the author credit on the site, if you are
>> interested in helping, let me know!). But I expect to have to code it
>> up myself. Can anyone see from these images what I am doing wrong? If
>> it will help if I give a code example, let me know and I will post
>> that...
>
> 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.

4) White pixels are appearing in the images.

Here is my code:

```
 ------------------------------------------------------------ -------
FUNCTION vmars, dem, img

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

;;; surface coordinate setup
sz = SIZE( dem )
minx = ( miny = 0 )
maxx = sz[ 1 ] - 1 & maxy = sz[ 2 ] - 1
maxz = MAX( dem, MIN=minz )
xs = [-minx / (maxx-minx), 1.0 / (maxx-minx) ]
ys = [-miny / (maxy-miny), 1.0 / (maxy-miny) ]
zs = [-minz / (maxz-minz), 1.0 / (maxz-minz) ] ; vert exag

;;; image -> surface coordinates
s = size( dem, /dim )
texcoords = fltarr( 2, s[0], s[1] )
texcoords[ 0, *, * ] = (findgen(s[0])#replicate(1,s[1])) / (s[0]-1)
texcoords[ 1, *, * ] = (replicate(1,s[1])#findgen(s[0])) / (s[0]-1)

;;; objects creation
window = obj_new( 'idlgrbuffer');, renderer=1 )
view = obj_new( 'idlgrview' , location=[0,-25] )
model = obj_new( 'idlgrmodel' )
rotate = obj_new( 'idlgrmodel' )
translate = obj_new( 'idlgrmodel' )
image = obj_new( 'idlgrimage', img )
surface = obj_new( 'idlgrsurface', $
                dem, $
                indgen(n_elements(dem[*,0])),$
                indgen(n_elements(dem[0,*])), $
                style=2, $
                color=[255,255,255], $
                texture_map=image, $
                texture_coord=texcoords, $
                shading=1, $
                xcoord_conv=xs, $
                ycoord_conv=ys, $
```

```
            zcoord_conv=zs/5., $
            dataz=dem )

;;; object heirarchy
view->add, translate
translate->add, rotate
rotate->add, model
model->add, surface

;;; view setup
rotate->rotate, [0,0,1], 45
rotate->rotate, [1,0,0], -50  ;;; down
translate->translate, 00.00, -0.8, 0.7

;;; display
window->draw, view
window->getProperty, image_data=surface
return, surface

END
```
