
Subject: Re: rotate and transpose images in xobjviewer

Posted by [Steven Houston](#) on Tue, 03 Apr 2007 10:06:10 GMT

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

Dick Jackson wrote:

> Hi vijay,

>

> "vijay" <vijayansiva@gmail.com> wrote in message

> news:1175343528.461497.166670@p15g2000hsd.googlegroups.com.. .

>> Hi, i am having a problem in rotating an image in the xobjviewer. the

>> problem is not with the rotate button but my image itself rotated and

>> transposed in the xobjviewer. So i want the image in the xobjviewer to

>> be transposed and rotated for 270 degrees. i can't able to do this

>> using the rotate command also. So tell me how to rotate and transpose

>> the image.

>

> The IDLgrImage object does not create a real 3D rotatable object, but what I

> think you want is a rectangular IDLgrPolygon object with an image as its texture

> map. (I'm guessing here, because you didn't give a lot of detail on what you had

> tried so far)

>

>

> ;; Get an image

>

> file = Filepath('rose.jpg', Subdir=['examples', 'data'])

> Read_JPEG, file, image

>

> ;; Make an IDLgrImage object

>

> olmage = Obj_New('IDLgrImage', image)

> XObjView, olmage, Title='olmage' ; Cannot really rotate the image

>

> ;; Make an IDLgrPolygon object with the image as its texture map

>

> olmage -> GetProperty, Dimensions=dims ; [width, height]

> oPoly1 = Obj_New('IDLgrPolygon', \$

> [[0,0,0],[dims[0],0,0],[dims[0],dims[1],0],[0,dims[1],0]], \$

> Color=[255,255,255], Texture_Map=olmage, \$

> Texture_Coord=[[0,0],[1,0],[1,1],[0,1]])

> XObjView, oPoly1, Title='oPoly1', XOffset=300

>

> ;; Try with no lighting to avoid change in shading when rotated

>

> oHiddenLight = Obj_New('IDLgrLight', Type=0, Color=[255,0,0], /Hide)

> XObjView, oPoly1, Title='oPoly1 with no lighting', XOffset=600, \$

> Stationary=oHiddenLight

>

> ;; Create polygon with coordinates for polygon rotated 270 degrees

```

>
> oPoly2 = Obj_New('IDLgrPolygon', $
>     [[0,dims[0],0],[0,0,0],[dims[1],0,0],[dims[1],dims[0],0]], $
>     Color=[255,255,255], Texture_Map=olImage, $
>     Texture_Coord=[[0,0],[1,0],[1,1],[0,1]])
> XObjView, oPoly2, Title='oPoly2 with no lighting', XOffset=600, YOffset=300, $
>     Stationary=oHiddenLight
>
>
> There are other subtle issues with image row order settings (the Order property
> to IDLgrImage) and resampling of image pixels (see the Texture_Map property of
> IDLgrPolygon), but is this what you were looking for?
>
> Cheers,
> -Dick
>
> --
> Dick Jackson Software Consulting      http://www.d-jackson.com
> Victoria, BC, Canada      +1-250-220-6117      dick@d-jackson.com
>
>

```

Hi,

As of IDL 6.2 you can apply 3D transforms to an IDLgrImage, but you need to set TRANSFORM_MODE=1 (the default is to use the 2D transform mode for backwards compatibility).

So in 6.2 and later you can do:

```

;; Get an image
file = Filepath('rose.jpg', Subdir=['examples', 'data'])
Read_JPEG, file, image

;; Make an IDLgrImage object
olImage = Obj_New('IDLgrImage', image, TRANSFORM_MODE=1)

;; Create a model to apply the 270 degree rotation
oModel = Obj_New('IDLgrModel')
oModel->Rotate, [0,0,1], 270
oModel->Add, olImage

XObjView, oModel, Title='olImage'

```

Cheers,
Steve.
