
Subject: IDLgrVolume rendering bug

Posted by [markb77](#) on Wed, 10 Jun 2015 11:56:03 GMT

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

hi,

I've run into what seems like a serious bug in the IDLgrVolume object (it was in fact Dick Jackson who uncovered this).

Here is the problem. When setting the LOCATION property of an IDLgrView object which contains the volume object, the volume object doesn't adjust to the new LOCATION. Rather, it remains rendered as if the LOCATION of the IDLgrView remains at [0,0].

I am rendering multiple IDLgrViews in an IDLgrScene object, and I am using the LOCATION property of the views to allow the user to move each view and its contents around the screen. All other graphics objects (lines, plots, images) seem to render normally and as expected when changing the LOCATION of the view. It is only the IDLgrVolume object that doesn't react.

I have written a short code to demonstrate the problem:

```
;-----
```

```
pro volume_render_bug
```

```
    ; determine the path to the image file
```

```
    imgfile = FILEPATH('rose.jpg', $  
        SUBDIRECTORY = ['examples', 'data'])
```

```
    READ_JPEG, imgfile, imgdat
```

```
    ; set up the volume data
```

```
    vdata = BYTARR(64,64,64)  
    FOR i=0,63 DO vdata[:,i,0:i] = i*2  
    vdata[5:15, 5:15, 5:55] = 128  
    vdata[45:55, 45:55, 5:15] = 255
```

```
    ; initialize window and view sizes
```

```
    winx = 650  
    winy = 750
```

```
    vol_viewx = 300  
    vol_viewy = 300
```

```
vol_view_dims = [vol_viewx,vol_viewy]
```

```
; Initialize graphics objects
```

```
; set up the image
```

```
oImage = OBJ_NEW('IDLgrImage', imgdat)
```

```
oImage -> GetProperty, DIMENSIONS = imgdims
```

```
imgx = imgdims[0]
```

```
imgy = imgdims[1]
```

```
img_model = OBJ_NEW('IDLgrModel')
```

```
img_model -> Add, oImage
```

```
oimgview = OBJ_NEW('IDLgrView', $  
    VIEWPLANE_RECT = [0.0, 0.0, imgx, imgy], $  
    LOCATION=[350,300], $  
    DIMENSIONS=imgdims, $  
    COLOR=[0,0,0])
```

```
oimgview -> Add, img_model
```

```
; set up the volume object
```

```
oVolume = OBJ_NEW('IDLgrVolume', vdata)
```

```
vol_datamodel = OBJ_NEW('IDLgrModel')
```

```
vol_rotationmodel = OBJ_NEW('IDLgrModel')
```

```
vol_datamodel -> Add, oVolume
```

```
vol_datamodel -> Translate, -32.0, -32.0, -32.0
```

```
vol_rotationmodel -> Add, vol_datamodel
```

```
ovolview = OBJ_NEW('IDLgrView', $  
    VIEWPLANE_RECT = [-50.0,-50.0,100.0,100.0], $  
    LOCATION=[0,0], $  
    DIMENSIONS=vol_view_dims, $  
    COLOR=[0,0,0], $  
    EYE = 65.0, $  
    ZCLIP = [64.0, -64.0])
```

ovolveview -> Add, vol_rotationmodel

; create a bounding box around the volume object

```
oboundingbox = obj_new('IDLgrModel')
```

```
bBoxThick = 1
```

```
bBoxColor = [255, 255, 255]
```

```
for x1=0, 1 do for y1=0, 1 do for z1=0, 1 do begin
```

```
  if x1 eq 0 then begin
```

```
    xLine = [[0, y1, z1], [1, y1, z1]]
```

```
    oboundingbox -> Add, $  
      obj_new('idlgrpolyline', $  
        xLine, $  
        thick=bBoxThick, $  
        Color=bBoxColor)
```

```
  endif
```

```
  if y1 eq 0 then begin
```

```
    yLine = [[x1, 0, z1], [x1, 1, z1]]
```

```
    oboundingbox -> Add, $  
      obj_new('idlgrpolyline', $  
        yLine, $  
        thick=bBoxThick, $  
        Color=bBoxColor)
```

```
  endif
```

```
  if z1 eq 0 then begin
```

```
    zLine = [[x1, y1, z1], [x1, y1, 1-z1]]
```

```
    oboundingbox -> Add, $  
      obj_new('idlgrpolyline', $  
        zLine, $  
        thick=bBoxThick, $  
        Color=bBoxColor)
```

```
  endif
```

endfor

vol_datamodel -> Add, oboundingbox
oboundingbox -> Scale, 64.0, 64.0, 64.0

; set up the on-screen text

msgtxt = ['Shifting the IDLgrView which contains the ' + \$
 'IDLgrImage works', \$
 'as expected.']

otext1 = obj_new('IDLgrText', msgtxt, \$
 LOCATIONS=[[0.1,0.9],[0.1,0.8]], \$
 COLOR=[255,255,255], \$
 HIDE=1)

msgtxt2 = ['There is a problem when shifting the ' + \$
 'IDLgrView which contains', \$
 'the IDLgrVolume object. The polylines shift ' + \$
 'with the IDLgrView', \$
 'but the volume does not move.']

otext2 = obj_new('IDLgrText', msgtxt2, \$
 LOCATIONS=[[0.1,0.6],[0.1,0.5],[0.1,0.4]], \$
 COLOR=[255,255,255], \$
 HIDE=1)

msgtxt3 = ['This appears to be a bug in the IDLgrVolume object']

otext3 = obj_new('IDLgrText', msgtxt3, \$
 LOCATIONS=[0.1,0.2], \$
 COLOR=[255,255,255], \$
 HIDE=1)

otextmodel = obj_new('IDLgrModel')

otextview = obj_new('IDLgrView', \$
 DIMENSIONS = [winx,200], \$
 LOCATION = [0,500], \$
 COLOR = [0,0,0], \$
 VIEWPLANE_RECT=[0.0,0.0,1.0,1.0])

otextmodel -> Add, otext1
otextmodel -> Add, otext2
otextmodel -> Add, otext3
otextview -> Add, otextmodel

```

; set up the scene

oscene = OBJ_NEW('IDLgrScene', COLOR=[0,0,0])

oscene -> Add, ovolview
oscene -> Add, oimgview
oscene -> Add, otextview

oWindow = OBJ_NEW('IDLgrWindow', $
    RETAIN = 0, $
    DIMENSIONS = [winx,winy], $
    TITLE = 'Volume render bug')

oWindow -> Erase, COLOR=[0,0,0]

oWindow -> Draw, oscene

wait, 0.5

otext1 -> SetProperty, HIDE=0

wait, 0.5

; move the image by changing the location property
; of the image view

nmoves = 200
movescale = 30.0

oimgview -> GetProperty, LOCATION=init_loc

for i = 0, nmoves-1 do begin

    dx = sin( (2.0*pi) * (i/float(nmoves-1)) ) * movescale
    dy = sin( (2.0*pi) * (i/float(nmoves-1)) ) * movescale

    oimgview -> SetProperty, $
        LOCATION=[init_loc[0]+dx,init_loc[1]+dy]

    oWindow -> Draw, oscene

endfor

wait, 0.5

otext2 -> SetProperty, HIDE=0

```

```

wait, 0.5

; rotate the volume

nrots = 200

for i = 0, nrots-1 do begin

    rot_incr = 360.0/nrots
    vol_rotationmodel -> Rotate, [1,1,0], rot_incr
    oWindow -> Draw, oscene

endfor

; rotate the volume while moving the view

nrots = 200
movedx = 300
movedy = 0

ovolview -> GetProperty, LOCATION=vol_init_loc
vx = vol_init_loc[0]
vy = vol_init_loc[1]

for i = 0, nrots-1 do begin

    rot_incr = 360.0/nrots
    move_x_incr = movedx/float(nrots)
    move_y_incr = movedy/float(nrots)

    vol_rotationmodel -> Rotate, [1,1,0], rot_incr

    ovolview-> SetProperty, $
        LOCATION = [vx+(i*move_x_incr), vy+(i*move_y_incr)]

    oWindow -> Draw, oscene

endfor

wait, 0.5

otext3 -> SetProperty, HIDE=0

wait, 0.5

oWindow -> Draw, oscene

```

end

;-----

Does anyone have any ideas about this?

thanks!

Mark
