Subject: Drawing 3D Objects properly
Posted by MKatz843 on Thu, 05 Sep 2002 07:49:58 GMT
View Forum Message <> Reply to Message

I'm very comfortable with object graphics in 2D. Now I've started branching out into the 3rd dimension. (No, I haven't yet seen the new object graphics book.)

Whatever I try, I can't seem to get solid-fill polygons to successfully "hide" or "block" another from view. With texture-mapped polygons it works. I've tried futzing with REJECT = [0,1,2], BOTTOM colors, you name it.

I'm just drawing 6-sided, rectangular, 3D boxes. By changing the handedness of the vectors I can make the surface seem to disappear in an undesirable way, but I can't get one surface to block another.

Any tips? This problem should be too simple for IDLgrVolume objects. I'd like to stick to IDLgrPolygon objects, if possible.

Thanks,

M. Katz

Subject: Re: Drawing 3D Objects properly
Posted by Rick Towler on Thu, 05 Sep 2002 15:51:30 GMT
View Forum Message <> Reply to Message

Can you post a simple example using xobjview to display?

-Rick

"M. Katz" <MKatz843@onebox.com> wrote in message news:4a097d6a.0209042349.663aca19@posting.google.com...
> I'm very comfortable with object graphics in 2D. Now I've started
> branching out into the 3rd dimension. (No, I haven't yet seen the new
> object graphics book.)
>
> Whatever I try, I can't seem to get solid-fill polygons to
> successfully "hide" or "block" another from view. With texture-mapped
> polygons it works. I've tried futzing with REJECT = [0,1,2], BOTTOM
> colors, you name it.
>
> I'm just drawing 6-sided, rectangular, 3D boxes. By changing the
> handedness of the vectors I can make the surface seem to disappear in
> an undesirable way, but I can't get one surface to block another.

>
> Any tips? This problem should be too simple for IDLgrVolume objects.
> I'd like to stick to IDLgrPolygon objects, if possible.
>
> Thanks,
>
> M. Katz

Subject: Re: Drawing 3D Objects properly
Posted by MKatz843 on Thu, 05 Sep 2002 18:08:21 GMT
View Forum Message <> Reply to Message

Thanks for the tip to use xobjview! In xobjview, I saw that the 3D objects behaved as they are supposed to. This led me to tweak all of the settings on my IDLgrView, until I found the culprit.

The PROJECTION keyword is supposed to change from orthogonal to a near/far view. In IDL 5.4 (at least), I see that it does more than that. If you run my simple side-by-side test script below, you'll see the issue.

With PROJECTION=1, one polygon successfully hides the one behind it. With PROJECTION=2, the red square (first drawn) *always* blocks the green square.

Run the test script below by copying it into a new file, name and save the file, and then .run the file. Move the mouse around in the IDL 0 window (the third window on screen, possibly on the right of the other two). You can control the rotation direction and velocity. Click in that window to quit.

I'm very interested to know if this is platform speciic or not. (I'm running IDL in Classic mode on MacOS 10.1.5)

M. Katz
(Script Follows)

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

; object-viewer test script demonstrating projection diffrerences

  ;--- create two identical windows
mywindow1 = obj_new('IDLgrWindow', quality=2, dimensions=[200,200], $
   LOCATION=[50,50])
mywindow2 = obj_new('IDLgrWindow', quality=2, dimensions=[200,200], $
   LOCATION=[50,300])
  ;--- create two views with different PROJECTION values
```

```
view1 = obj_new('IDLgrView',EYE=1000, zclip=[999,-999], $
   color=[0,0,0], VIEWPLANE_RECT = [-1,-1,2,2]*2, PROJECTION=1)
view2 = obj_new('IDLgrView',EYE=1000, zclip=[999,-999], $
   color=[0,0,0], VIEWPLANE_RECT = [-1,-1,2,2]*2, PROJECTION=2)

  ;--- Create two identical models: A red and a green square with
different z values
model1 = obj_new('IDLgrModel')
model1 -> Add, obj_new('IDLgrPolygon', [-1,1,1,-1], [1,1,-1,-1], $
  [0,0,0,0]-0.5, COLOR=[200,50,50])
model1 -> Add, obj_new('IDLgrPolygon', [-1,1,1,-1], [1,1,-1,-1], $
  [0,0,0,0]+0.5, COLOR=[50,200,50])

model2 = obj_new('IDLgrModel')
model2 -> Add, obj_new('IDLgrPolygon', [-1,1,1,-1], [1,1,-1,-1], $
  [0,0,0,0]-0.5, COLOR=[200,50,50])
model2 -> Add, obj_new('IDLgrPolygon', [-1,1,1,-1], [1,1,-1,-1], $
  [0,0,0,0]+0.5, COLOR=[50,200,50])

view1 -> Add, model1
view2 -> Add, model2

  ;--- Allow the user to manipulate the graphics as they are displayed
window, xsize=200, ysize=200

repeat begin
  cursor, xc, yc, /normal, /nowait

  model1 -> Rotate, [0,1,0], (xc-0.5)*5
  model1 -> Rotate, [1,0,0], (yc-0.5)*5
  model2 -> Rotate, [0,1,0], (xc-0.5)*5
  model2 -> Rotate, [1,0,0], (yc-0.5)*5

  mywindow1 -> Draw, view1
  mywindow2 -> Draw, view2
endrep until (!mouse.button GT 0)

end
```

Subject: Re: Drawing 3D Objects properly
Posted by Rick Towler on Thu, 05 Sep 2002 18:33:35 GMT
View Forum Message <> Reply to Message

Yes, xobjview is a *very* useful tool. Too bad it doesn't get any ink.

Your problem must be a Mac thing? It looks good to me with either projection. Attached is an image showing your programs output.

Nix that image, my news server won't let me post it (sheesh, it is
only 8k). But what I did see looked correct. I can send the
image off list if you wish.

{x86 Win32 Windows Microsoft Windows 5.5 Aug 28 2001 32 64}

-Rick

"M. Katz" <MKatz843@onebox.com> wrote in message
news:4a097d6a.0209051008.6857593@posting.google.com...
> Thanks for the tip to use xobjview! In xobjview, I saw that the 3D
> objects behaved as they are supposed to. This led me to tweak all of
> the settings on my IDLgrView, until I found the culprit.
>
> The PROJECTION keyword is supposed to change from orthogonal to a
> near/far view. In IDL 5.4 (at least), I see that it does more than
> that. If you run my simple side-by-side test script below, you'll see
> the issue.
>
> With PROJECTION=1, one polygon successfully hides the one behind it.
> With PROJECTION=2, the red square (first drawn) *always* blocks the
> green square.
>
> Run the test script below by copying it into a new file, name and save
> the file, and then .run the file. Move the mouse around in the IDL 0
> window (the third window on screen, possibly on the right of the other
> two). You can control the rotation direction and velocity. Click in
> that window to quit.
>
> I'm very interested to know if this is platform speciic or not. (I'm
> running IDL in Classic mode on MacOS 10.1.5)
>
> M. Katz
> (Script Follows)
>
> ;---
>
> ; object-viewer test script demonstrating projection diffrerences
>
> ;--- create two identical windows
> mywindow1 = obj_new('IDLgrWindow', quality=2, dimensions=[200,200], $
> LOCATION=[50,50])
> mywindow2 = obj_new('IDLgrWindow', quality=2, dimensions=[200,200], $
> LOCATION=[50,300])
> ;--- create two views with different PROJECTION values
> view1 = obj_new('IDLgrView',EYE=1000, zclip=[999,-999], $

```
>     color=[0,0,0], VIEWPLANE_RECT = [-1,-1,2,2]*2, PROJECTION=1)
>  view2 = obj_new('IDLgrView',EYE=1000, zclip=[999,-999], $
>     color=[0,0,0], VIEWPLANE_RECT = [-1,-1,2,2]*2, PROJECTION=2)
>
>    ;--- Create two identical models: A red and a green square with
>  different z values
>  model1 = obj_new('IDLgrModel')
>  model1 -> Add, obj_new('IDLgrPolygon', [-1,1,1,-1], [1,1,-1,-1], $
>    [0,0,0,0]-0.5, COLOR=[200,50,50])
>  model1 -> Add, obj_new('IDLgrPolygon', [-1,1,1,-1], [1,1,-1,-1], $
>    [0,0,0,0]+0.5, COLOR=[50,200,50])
>
>  model2 = obj_new('IDLgrModel')
>  model2 -> Add, obj_new('IDLgrPolygon', [-1,1,1,-1], [1,1,-1,-1], $
>    [0,0,0,0]-0.5, COLOR=[200,50,50])
>  model2 -> Add, obj_new('IDLgrPolygon', [-1,1,1,-1], [1,1,-1,-1], $
>    [0,0,0,0]+0.5, COLOR=[50,200,50])
>
>  view1 -> Add, model1
>  view2 -> Add, model2
>
>    ;--- Allow the user to manipulate the graphics as they are displayed
>  window, xsize=200, ysize=200
>
>  repeat begin
>    cursor, xc, yc, /normal, /nowait
>
>    model1 -> Rotate, [0,1,0], (xc-0.5)*5
>    model1 -> Rotate, [1,0,0], (yc-0.5)*5
>    model2 -> Rotate, [0,1,0], (xc-0.5)*5
>    model2 -> Rotate, [1,0,0], (yc-0.5)*5
>
>    mywindow1 -> Draw, view1
>    mywindow2 -> Draw, view2
>  endrep until (!mouse.button GT 0)
>
>  end
```

Subject: Re: Drawing 3D Objects properly
Posted by Karl Schultz on Thu, 05 Sep 2002 21:07:28 GMT
View Forum Message <> Reply to Message

Try reducing the ZCLIP range..

If I made ZCLIP=[999,-999] / 100, the problem went away.

Here's why:

The hardware Z buffer has limited precision. Many Z buffers have only 16 bits per pixel and it varies with the implementation. This may explain why the problem appears worse on some systems as compared to others. In addition, if you are on a machine that supports OpenGL, you may see some differences between the hardware and software renderers, because the Z buffer depths may be different between the OpenGL on your machine and the IDL software renderer (Mesa).

When you draw a unit polygon like this that takes up about 2 units in a Z buffer range of [-999, 999], you are only using 1/1000th of the Z buffer. So, if the Z buffer is 16-bit, you are only going to be using 64 of the possible 64K Z buffer values. This results in essentially 64 discreet "depth" values used while drawing these two polygons. If the polygons get close enough in Z as you spin them around, the Z values of some of the pixels after the view projection will map to the same Z buffer value. When there is a tie in Z, the drawing order becomes the deciding factor.

The graphics system (OpenGL) maps the Z part of the viewport to the Z buffer differently, depending on the PROJECTION, which may account for the difference there. When the projection is perspective, the ZCLIP-to-Zbuffer mapping is not linear. This means that you might be able to resolve the polygons at one end of the ZCLIP space and not the other. This explains why you were not able to resolve the polygons in the perspective projection.

If you were to make the ZCLIP values really big, you'd start to see stitching effects in both projections.

In general, you don't want to set your ZCLIP values to such large numbers. It is better to get them closer to the actual extents of your model.

This problem did not show up in XOBJVIEW because XOBJVIEW walks the graphics tree you provide to determine the Z extents of your data.

Try setting ZCLIP to [10,-10].

Hope this helps,
Karl

Subject: Re: Drawing 3D Objects properly
Posted by David Fanning on Fri, 06 Sep 2002 01:29:41 GMT
View Forum Message <> Reply to Message

Rick Towler (rtowler@u.washington.edu) writes:

> Your problem must be a Mac thing? It looks good to me with either
> projection. Attached is an image showing your programs output.

> Nix that image, my news server won't let me post it (sheesh, it is
> only 8k). But what I did see looked correct. I can send the
> image off list if you wish.
>
> {x86 Win32 Windows Microsoft Windows 5.5 Aug 28 2001 32 64}

Humm. That's odd. It looks wrong to me in both IDL 5.4
and IDL 5.5 on my Windows machine. Are you *sure* it looks
correct. In the second window, the red plane never goes
behind the green plane.

Cheers,

David

--
David W. Fanning, Ph.D.
Fanning Software Consulting, Inc.
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Drawing 3D Objects properly
Posted by Rick Towler on Fri, 06 Sep 2002 21:56:14 GMT
View Forum Message <> Reply to Message

"David Fanning" <david@dfanning.com> wrote ...
> Rick Towler (rtowler@u.washington.edu) writes:
>
>> Your problem must be a Mac thing? It looks good to me with either
>> projection. Attached is an image showing your programs output.
>> Nix that image, my news server won't let me post it (sheesh, it is
>> only 8k). But what I did see looked correct. I can send the
>> image off list if you wish.
>>
>> {x86 Win32 Windows Microsoft Windows 5.5 Aug 28 2001 32 64}
>
> Humm. That's odd. It looks wrong to me in both IDL 5.4
> and IDL 5.5 on my Windows machine. Are you *sure* it looks
> correct. In the second window, the red plane never goes
> behind the green plane.

Sure as sure can be. I just looked at the evidence again. The top and
bottom windows look identical.

Last I knew you had a nVidia Quadro2 based video card. I have an nVidia
card too so that is a bit odd that you would see something different. Maybe

nVidia has been doing some work with their z-buffer. I am running the 30.82
drivers with a GF3.

-Rick

Subject: Re: Drawing 3D Objects properly
Posted by David Fanning on Sat, 07 Sep 2002 02:29:48 GMT
View Forum Message <> Reply to Message

Rick Towler (rtowler@u.washington.edu) writes:

> Sure as sure can be. I just looked at the evidence again. The top and
> bottom windows look identical.
>
> Last I knew you had a nVidia Quadro2 based video card. I have an nVidia
> card too so that is a bit odd that you would see something different. Maybe
> nVidia has been doing some work with their z-buffer. I am running the 30.82
> drivers with a GF3.

Ah, I was running with the software renderer, not the
hardware. With hardware, everything renders correctly.
(Which, of course, is just the opposite of what usually
happens.) The ZCLIP tip offered by Karl also clears
up the problem for both hardware and software rendering.

Cheers,

David
--
David W. Fanning, Ph.D.
Fanning Software Consulting, Inc.
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155
