Subject: z-buffer
Posted by sjang on Mon, 05 Aug 1996 07:00:00 GMT
View Forum Message <> Reply to Message

We are trying to display a three-dimensional image such that we can view it from any angle we choose. From our experience, we thought that z-buffer mode might be the best way to accomplish this task; however, there's very little good documentation on z-buffer in the idl manuals. Could someone perhaps shed some light on this matter? For instance, is z-buffer the best way to go? Or is there a better or more efficent alternative? Any help would be greatly appreciated.

Sunmee Jang and Brad Fisher

Subject: Re: Z-Buffer
Posted by davidf on Mon, 10 Feb 1997 08:00:00 GMT
View Forum Message <> Reply to Message

John-David Smith <jdsmith@astrosun.tn.cornell.edu> writes:

> Dear Experts of the Buffer Z;
>
> I am trying to overlay an image created in the z-buffer with axes
> produced outside the z-buffer (to avoid pixelation and make nice-looking
> output) in a postscript image. The problem is, if the entire postscript
> region isn't used (e.g. to have multiple plots on a page), then the 3-d axes
> do not overlay the z-buffer images correctly. I could not find a way to
> map the 3-d position coordinates to the 2-d postscript coordinates for
> sub-regions of the postscript page.
>
> Any suggestions?

Here is a bit of IDL code that illustrates the problem John-David is having. You will notice that the axes don't line up, even though the same code is used to display the surface both in the Z-buffer and on the display.

```
  TVLCT, [255, 0], [255, 255], [0,0], 1
  data = DIST(40,40)
  thisDevice = !D.NAME
  SET_PLOT, 'Z'
  DEVICE, Set_Resolution=[300,300]
  SURFACE, data, Color=1
  picture = TVRD()
  SET_PLOT, thisDevice
  WINDOW, XSize=300, YSize=300
  TV, picture
```

```
   SURFACE, data, /NoErase, /NoData, Color=2
```

I've sent John-David a solution to this problem via private e-mail and have sworn him to secrecy so we can have a little diversion on this newsgroup.

This problem illustrates one of the deepest mysteries about IDL that I know. I don't think there are many IDL programmers who can solve this problem. If you can, you get an automatic invite to the IDL Expert Programmers Convention. So here is the contest.

I have a brand new copy of the Michael Dorris' novel A Yellow Raft in Blue Water (a must-read book if you haven't read it yet). I'll give it to the first person who posts an IDL program that draws a surface in the Z-buffer and labels the axes in the proper locations on the display. Put the time you post the result in your post, so I don't have to worry about how long the news machines take to get it to me.

If we don't have any winners, I'll post my answer next week.
(It will probably be wrong!)

All right. On your marks, get set, go! (And no cheating!) :-)

David Fanning, Ph.D.
Fanning Software Consulting
2642 Bradbury Court, Fort Collins, CO 80521
Phone: 970-221-0438   Fax: 970-221-4762
E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com

Subject: Re: Z-Buffer
Posted by Robert.M.Candey on Tue, 11 Feb 1997 08:00:00 GMT

View Forum Message <> Reply to Message

In article <davidf-ya023080001002972203460001@news.frii.com>,
davidf@dfanning.com (David Fanning) wrote:

> Here is a bit of IDL code that illustrates the problem John-David
> is having. You will notice that the axes don't line up, even though
> the same code is used to display the surface both in the Z-buffer
> and on the display.
>
> ```
>    TVLCT, [255, 0], [255, 255], [0,0], 1
> ```

```
>     data = DIST(40,40)
>     thisDevice = !D.NAME
>     SET_PLOT, 'Z'
>     DEVICE, Set_Resolution=[300,300]
>     SURFACE, data, Color=1
>     picture = TVRD()
>     SET_PLOT, thisDevice
>     WINDOW, XSize=300, YSize=300
>     TV, picture
>     SURFACE, data, /NoErase, /NoData, Color=2
>
```

> I've sent John-David a solution to this problem via private
> e-mail and have sworn him to secrecy so we can have a little
> diversion on this newsgroup.
>
> This problem illustrates one of the deepest mysteries about
> IDL that I know. I don't think there are many IDL programmers
> who can solve this problem. If you can, you get an automatic
> invite to the IDL Expert Programmers Convention. So here is
> the contest.
> ...

Ah, the answer is simple: add ",set_char=[6,10]" to the device line. This
defines the same character size as X, so the margins become the same size.
(submitted 1997 Feb 11 20:50 EST)

--
Robert.M.Candey@gsfc.nasa.gov
NASA Goddard Space Flight Center, Code 632
Greenbelt, MD 20771 USA 1-301-286-6707

Subject: Re: Z-Buffer
Posted by davidf on Tue, 11 Feb 1997 08:00:00 GMT

View Forum Message <> Reply to Message

JD Smith <jdsmith@astrosun.tn.cornell.edu> writes:

> Perhaps I misphrased my question...
>
>   I don't mean to break up the contest, but my real question is as follows:
> Suppose you create an image in the Z-buffer, but put no axes on it. This
image
> is read from the buffer into a variable. Since I had already discovered Mr.
> Savoie's solution to the character size problem (which is what is actually
> causing the misalignment of axes in Mr. Fanning's posited conundrum), set the
> character size appropriately. And now the fun part.... Switch to the
> postscript device, display the z-buf image in a position you'd like, and *now*

> put axes on top. The reasoning for this is obvious: unless you'd like to set
> your z-buffer resolution to that of your postscript device, the text of
the axes
> will not survive that foul temptress the z-buffer with even close to
> satisfactory quality.

I'm going to work on JDs problem some more, but I'm calling the whole contest off. Have you ever noticed that you learn more about IDL when you *think* you know what you are doing, but you don't?

I thought about sending a book to everyone who submitted a better solution than mine, but it turns out I don't have that many books in my library! :-)

Anyway, I've learned that there are at *least* three or four different ways to solve my original problem. I'll announce the winners shortly.

David

David Fanning, Ph.D.
Fanning Software Consulting
2642 Bradbury Court, Fort Collins, CO 80521
Phone: 970-221-0438 Fax: 970-221-4762
E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com

Subject: Re: Z-Buffer
Posted by savoie on Tue, 11 Feb 1997 08:00:00 GMT

View Forum Message <> Reply to Message

> I have a brand new copy of the Michael Dorris' novel A Yellow
> Raft in Blue Water (a must-read book if you haven't read it
> yet). I'll give it to the first person who posts an IDL program
> that draws a surface in the Z-buffer and labels the axes in the
> proper locations on the display. Put the time you post the result
> in your post, so I don't have to worry about how long the news
> machines take to get it to me.
>
> If we don't have any winners, I'll post my answer next week.
> (It will probably be wrong!)

I don't know if you wanted to know how long it took (~1h to find all the variables that changed when I switched from x to Z buffer, but I've never

used the Z buffer before, so that doesn't seem that bad). But here's one
solution using the program that you used first.

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
pro zbuffit
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

  tvlct, [255,0], [255,255], [0,0], 1

  data =  dist(40,40)
  thisdev =  !d.name
  dvar =  !d
  set_plot, 'Z'
  device, set_resolution=[300,300]
  device, set_character_size= [dvar.x_ch_size,dvar.y_ch_size]
  surface, data, color=1
  picture = tvrd()
  set_plot, thisdev
  window, xs=300, ys=300, /free
  tv, picture
  surface, data, /nodata, /noerase, color=2

END
```

This works on my machine.
cheers,
Matt
entry time
Tue Feb 11 10:10:19 MST 1997

--
```
Matthew H. Savoie                    Systems Technology Associates
Systems Analyst                   Supporting NOAA Profiler Network
ph. 303.497.6642                    Demonstration Division/FSL/ERL
mailto: savoie@fsl.noaa.gov  <URL:http://www-dd.fsl.noaa.gov/online.html>
```

Subject: Re: Z-Buffer
Posted by J.D. Smith on Wed, 12 Feb 1997 08:00:00 GMT

View Forum Message <> Reply to Message

G.M.Sigut wrote:
>

> The following might be a (beginning of a) solution:

>
> I took the example "as is", added some test data generation, made it
> work and then commented. I doubled the example to have two drawings
> under each other, which is (I presume) the goal. I did NOT try to
> make the whole thing look nice, but just made it work. The original
> text starts with an offset, practically all my additions start in

That solution did the trick! I modified it slightly however -- instead of forcing the PS page to conform to the dimensions of the Z-buffer, I set the resolution of the z-buffer to conform to the ratio of my output page.

The key to the solution was, however, that, inside the z-buffer, my plotting routine obeyed to !P.POSITION variable, since it used shade_surf only. This is a fine solution to this problem (and I wish I had thought of it), but it doesn't address the following larger issue: suppose what you needed to put into postscript was *actually* generated from some image command, and not a plotting command. For a specific example, suppose you had a galaxy survey with three dimension (ra, dec, and redshift) as an image, and you wanted to overlay axes. Getting the axes registered to the image is difficult, it seems to me, because of this dichotomy between plot and image positioning paradigms. But maybe I'm just missing something obvious. Anyway, some food for thought.

Thanks for the help,

JD

Subject: Re: Z-Buffer
Posted by sigut on Wed, 12 Feb 1997 08:00:00 GMT

View Forum Message <> Reply to Message

In article <33010900.41C67EA6@astrosun.tn.cornell.edu> John-David Smith <jdsmith@astrosun.tn.cornell.edu> writes:

> Here's some summarizing code (for those whose decompose the written word more
> readily into code fragments and blocks than sentences and paragraphs):

The following might be a (beginning of a) solution:

I took the example "as is", added some test data generation, made it
work and then commented. I doubled the example to have two drawings
under each other, which is (I presume) the goal. I did NOT try to
make the whole thing look nice, but just made it work. The original
text starts with an offset, practically all my additions start in col.1

Don't even start reading, if you don't REALLY want to know!

```
e0=findgen(21)
e1=fltarr(21,21)
e2=e1
for i=0,20 do begin &$
  e1(i,*)=e0 &$
  e2(*,i)=e0 &$
endfor
zmt0=dist(21)
zmt=zmt0/max(zmt0)*1.3
zm1=-zmt+1.3
zm2=zmt0*0+.65
; that was the test data generation
   dev=!D
   set_plot,'Z'
   !P.BACKGROUND=!D.N_COLORS-1 ; to get white background
   device,set_character_size=[dev.x_ch_size,dev.y_ch_size]
!p.position=[.1,.55,.9,.9]
; since you want to have the shape and the axes in the same place
; you will have to work with !p.position
   ;; put 3 surfs atop eachother ... the z's and e's are concocted elsewhere
   shade_surf,zmt,e1,e2,shades=bytscl(zmt,TOP=!d.table_size), $
    charsi=2.25,/save,az=20,zrange=[0,1.3],xst=4,yst=4,zst=4
tr1=!p.t
; better save your current trafo
   shade_surf,zm1,e1,e2,shades=zm1*0B+!d.table_size/3,/T3D, $
    xst=4,yst=4,zst=4,charsi=2.25,/NOERASE,zrange=[0,1.3]
    shade_surf,zm2,e1,e2,shades=zmt*0B+!d.table_size/4,/NOERASE, /T3D, $
    xstyle=4,ystyle=4,zstyle=4,charsi=2.25,zrange=[0,1.3]
   ;; get the pixture out
;  a=tvrd()
; don't save the picture, you are not done yet
!p.position=[.1,.1,.9,.45]
; here the lower picture
   ;; put 3 surfs atop eachother ... the z's and e's are concocted elsewhere
   shade_surf,zmt,e1,e2,shades=bytscl(zmt,TOP=!d.table_size), $
     charsi=2.25,/save,az=20,zrange=[0,1.3],xst=4,yst=4,zst=4,/no erase
;   charsi=2.25,/save,az=20,zrange=[0,1.3],xst=4,yst=4,zst=4
; had to add /noerase (both pictures on 1 frame)
tr2=!p.t
; and save the second trafo
   shade_surf,zm1,e1,e2,shades=zm1*0B+!d.table_size/3,/T3D, $
    xst=4,yst=4,zst=4,charsi=2.25,/NOERASE,zrange=[0,1.3]
    shade_surf,zm2,e1,e2,shades=zmt*0B+!d.table_size/4,/NOERASE, /T3D, $
    xstyle=4,ystyle=4,zstyle=4,charsi=2.25,zrange=[0,1.3]
   ;; get the pixture out
   a=tvrd()
```

```
ratio=float(!d.y_size)/!d.x_size
; have to keep the current frame ratio
set_plot,'ps'
; I think the set_plot was forgotten in the original
;  device,BITS=8, xsize=8.5,ysize=11.,xoffset=0.,yoffset=0.,/inches
   device,BITS=8, xsize=7.,ysize=7.*ratio,xoffset=1.,yoffset=1.,/inches
; I adjusted the xsize and offsets to fit A4 paper format
; ysize is NOT free, but has to be adjusted to the proportions used
; in the z-buffer
;  !P.MULTI=[0,1,3,0,0]
; You can't use that - or rather I can't use that in this context :-)
;  tv,a,.5,5,xsize=5,ysize=5,/inches
   tv,a
; you cannot put your picture SOMEWHERE, but only where it was (i.e. 0,0)
; also you can't adjust the size as you please (well...probably you COULD
; if you would coordinate it with the "ratio", but..)
   ;; now add the axes....
!p.position=[.1,.55,.9,.9]
!p.t=tr1
; restore the !p.position and transformation for the 1st picture
   shade_surf,zmt,e1,e2,/T3D,xtitle='e!d1',ytitle='e!d2', $
    ztitle='Normalized Merit',/NODATA,/NOERASE, $
    TITLE='Ritchey-Chretian Optimization',charsi=2.25,zrange=[0,1.3]
!p.position=[.1,.1,.9,.45]
!p.t=tr2
; restore the !p.position and transformation for the 2nd picture
   shade_surf,zmt,e1,e2,/T3D,xtitle='e!d1',ytitle='e!d2', $
    ztitle='Normalized Merit',/NODATA,/NOERASE, $
    TITLE='Ritchey-Chretian Optimization',charsi=2.25,zrange=[0,1.3]
   device,/close
   set_plot,'X'
; ... and viola! (TM)
```

Greetings to everybody who made it through,

George   ;-)

```
--
  ------------------------------------------------------------ ------------------
 George M.Sigut,  ETH Informatikdienste, Anwenderunterstuetzung, CH-8092 Zurich
 Swiss Federal Institute of Technology,    Computing Services,     User Support
 email: sigut@awu.id.ethz.ch     Phone: +41 1 632 5763      Fax: +41 1 632 1312
 >>>>>>>>> in case of email problems send the mail to "sigut@acm.org" <<<<<<<<<
  ------------------------------------------------------------ ------------------
```

Subject: Re: Z-Buffer
Posted by sigut on Thu, 13 Feb 1997 08:00:00 GMT

In article <330206B8.41C67EA6@astrosun.tn.cornell.edu>
John-David Smith <jdsmith@astrosun.tn.cornell.edu> writes:

> ... but it doesn't address the following larger issue:
> suppose what you needed to put into postscript was *actually* generated
> from some image command, and not a plotting command. For a specific
> example, suppose you had a galaxy survey with three dimension (ra, dec,
> and redshift) as an image, and you wanted to overlay axes. Getting the
> axes registered to the image is difficult, it seems to me, because of
> this dichotomy between plot and image positioning paradigms.

In 2d you can do something like:

```
;
;  raster.script
;

@get_stars.script  ; this reads in a byte array (480,512)

a=intarr(2,2)
!p.title='Feb 21 00:54:38 1992'

set_plot,'ps'

device,xoffset=4

if float(!d.x_size)/!d.y_size lt float(480)/512 then $
   !p.region=[0,0,1,512.*(float(!d.x_size)/480./float(!d.y_size ))] $
else $
  !p.region=[0,0,  480.*(float(!d.y_size)/512./float(!d.x_size)),1]

tv,b

plot,a,/nodata,/noerase,xrange=[0,1],yrange=[0,1], $
xmargin=[0,0],ymargin=[0,0],ticklen=-.02,xstyle=8

device,/close
set_plot,'x'
```

For 3-dim I just don't see quite clearly how your image is defined,
what axes do you want and where do you want them. How about an email
with details?

George
--
  -- ------------------
 George M.Sigut, ETH Informatikdienste, Anwenderunterstuetzung, CH-8092 Zurich

Swiss Federal Institute of Technology,   Computing Services,   User Support
email: sigut@awu.id.ethz.ch   Phone: +41 1 632 5763   Fax: +41 1 632 1312
>>>>>>>>> in case of email problems send the mail to "sigut@acm.org" <<<<<<<<<
-- ------------------

Subject: Re: Z-Buffer
Posted by davidf on Wed, 27 Oct 1999 07:00:00 GMT
View Forum Message <> Reply to Message

Haje Korth (hkorth@lanl.gov) writes:

> Is it possible to use the Z-Buffer device and still get decent output
> quality for characters when an image of the buffer (tvrd) is dumped to
> the Postscript device. Basically I want to use the Z-Buffer image with
> axes rendered in Postscript. Is this possible, or do I ask for too much
> here?

All things are "possible" in IDL, and sometimes they
are even "likely", if you know what you are doing. :-)

I've certainly done what you are attempting many times.
But I should warn you about one little peculiarity in
IDL that will absolutely drive you crazy! The character
size in the Z-graphics buffer is just *slightly* different
from the character size on your display. So aligning
axes with data can be an exercise in total frustration
is you are as anally-retentive about how things look as
I am.

I've learned that *anytime* I try to combine
Z-buffer graphics with regular graphics what it is
absolutely critical to set the Charsize=1.0 keyword
for *ALL* graphics commands carried out in the Z-buffer.

Cheers,

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