Subject: DirectColor on linux
Posted by doosh on Mon, 22 Nov 1999 08:00:00 GMT
View Forum Message <> Reply to Message

[This message was originally posted to comp.lang.idl; thanks for the redirect]

One of the professors I support has a procedure he's using on Solaris with IDL that he wants to move over to a linux machine. However, there appears to be some fundamental differences in the way Solaris handles colors, and the procedure isn't working.

Basically, what the procedure does is use direct color to change the color table, thereby changing the colors of the images in window 0. This is useful in astronomy because it allows you to highlight different parts of the spectrum. So if you diddle the red from 255 down to 0, all the red will be drained out of the image. I've seen this work on a Solaris machine. I am trying to do it on a Linux machine with a Diamond Viper 770 (32MB) video card.

The problem is, it doesn't work. I can set device,direct=24 under XFree86 or Accelerated X, and IDL reports that it's using direct color, but running the diddle procedure (attached below) doesn't change the colors in window 0.

I don't really understand the issues involved; is it something IDL needs to support, or the X server, or the video card? Is it possible to do this with IDL on Linux at all?
-Tom

```
;***** CARLSTRETCH*****************************************
pro carlstretch, r, g, b, low, high, gamma
;like IDL's 'stretch', but does color independently.
common colors, r_orig, g_orig, b_orig, r_curr, g_curr, b_curr
nc = !d.table_size      ;# of colors entries in device

;help, r, g, b, low, high, gamma
;return
slope = 1. / (float(high) - float(low))         ;Range of 0 to 1.
intercept = -slope * float(low)
p = findgen(nc) * slope + intercept > 0.0
p = long(nc * (p ^ gamma)) < 255

if (r ne 0) then r_curr = r_orig[p]
if (g ne 0) then g_curr = g_orig[p]
if (b ne 0) then b_curr = b_orig[p]

;tempwindow = !d.window
```

```
;wset, 0
;plot, r_curr, xstyle=1, xrange=[0,255], ystyle=1, yrange=[0,255], color=255
;oplot, g_curr, color=255l*256l
;oplot, b_curr, color=255l*256l*256l
;wset, tempwindow

tvlct, r_curr, g_curr, b_curr
return
end

;***** DIDDLE ****************************************
pro diddle, colors, lo2, hi2, gamma2
;diddles gamma and stretch of images in 3 channels independently or in all.
;which color depends on colors. r, g, b or any combination.

;BEGIN WITH THE INPUT STRETCH PARAMETERS IF THEY HAVE BEEN ENTERED
;OTHERWISE, USE THE DEFAULT VALUES.

;FIRST THE COLORS TO PROCESS...
common colors, r_orig, g_orig, b_orig, r_curr, g_curr, b_curr
if (n_params() eq 0) then colors='rgb'
r=0l & g=0l & b=0l
if (strpos( colors, 'r') ne -1) then r=255l
if (strpos( colors, 'g') ne -1) then g=255l
if (strpos( colors, 'b') ne -1) then b=255l
colorout = r + 256l*( g + 256l*b)
print, r, g, b, colorout

;MAKE SURE A COLOR TABLE HAS BEEN LOADED...
catch, err_in
if (err_in ne 0) then begin
loadct, 0
endif
txtst = r_curr(0)

;THEN THE OTHER PARAMETERS...
catch, err_in
if (err_in ne 0) then begin
lo2 = 0
endif
lo1 = float(lo2)/255.

catch, err_in
if (err_in ne 0) then begin
hi2 = 255
endif
hi1 = float(hi2)/255.
```

```
catch, err_in
if (err_in ne 0) then begin
gamma2 = 1.0
endif
gamma1 = 0.5*( 1. + alog10( gamma2))

catch, err_in, /cancel
;lo1 = 0.
;hi1 = 1.
;gamma1 = 0.5
lo2 = byte( lo1 * 255)
hi2 = byte( hi1 * 255)
gamma2 = 0.1*(100.^gamma1)
;print, colors, lo1, hi1, gamma1
;print, lo2, hi2, gamma2

;SAVE THE ORIGINAL WINDOW NUMBER SO THAT WE GO BACK TO IT UPON RETURN...
windownr = !d.window
print, 'original window = ', windownr

;DEFINE AND POPULATE THE DIDDLE WINDOW...
window, xsize=200, ysize=100, xpos=50, ypos=50, /free, retain=2
windowdiddle = !d.window
plot, [0,1], [0,1], xrange=[0, 1], yrange=[0,1], /nodata, $
 xstyle=1, ystyle=1, position=[.05, .05, .95, .95], color=colorout
xyouts, .1, .2, lo2, /normal, charsize=1.5, color=colorout
xyouts, .1, .5, hi2, /normal, charsize=1.5, color=colorout
xyouts, .1, .8, gamma2, /normal, charsize=1.5, color=colorout
lo2old = lo2
hi2old = hi2
gamma2old = gamma2

!err = 2
print, 'left button controls min, mid button gamma, rght button max'
print, 'the horixontal position of the cursor gives the value '
print, 'any two buttons simultaneously leaves the routine.
print, ' '
print, 'STARTING low, high, gamma = ', lo2, hi2, gamma2, $
 '; color = ', colors

beginagn:

;print, r, b, g, lo2, hi2, gamma2
carlstretch, r, g, b, lo2, hi2, gamma2

;stop
cursor, xx, yy, 0, /data
```

```
;print, !err
if ( (!err ne 0) and (!err ne 1) and (!err ne 2) and (!err ne 4)) then begin
wset, windownr
wdelete, windowdiddle
print, 'return to window number ', windownr
print, 'ENDING   low, high, gamma = ', lo2, hi2, gamma2
;stop
return
endif

xx = (0. > xx) < 1.0

if (!err eq 1) then begin
lo2 = byte( xx * 255)
carlstretch, r, g, b, lo2, hi2, gamma2
endif

if (!err eq 4) then begin
hi2 = byte( xx * 255)
carlstretch,  r, g, b, lo2, hi2, gamma2
endif

if (!err eq 2) then begin
gamma2 = 0.1*(100.^xx)
carlstretch,  r, g, b, lo2, hi2, gamma2
endif

xyouts, .1, .2, lo2old, /normal, charsize=1.5, color=0
xyouts, .1, .5, hi2old, /normal, charsize=1.5, color=0
xyouts, .1, .8, gamma2old, /normal, charsize=1.5, color=0
xyouts, .1, .2, lo2, /normal, charsize=1.5, color=colorout
xyouts, .1, .5, hi2, /normal, charsize=1.5, color=colorout
xyouts, .1, .8, gamma2, /normal, charsize=1.5, color=colorout
lo2old = lo2
hi2old = hi2
gamma2old = gamma2
wait, 0.01
goto, beginagn

return
end
```

Subject: Re: DirectColor on linux
Posted by davidf on Tue, 23 Nov 1999 08:00:00 GMT
View Forum Message <> Reply to Message

Vincent Favre-Nicolin (favre@polycnrs-gre.fr) writes:

> Well, I woul like very much not using XOR, but then
> I would have to redisplay the image (they can be up to 5000x3500,float)
> each time I add some text on it. A bit time and memory-consuming...

Well, not really. With Device Copy you can repair just
that portion of the window that you drew something on.
That is usually an extremely small region (e.g., a
selection box, maybe). This is absolutely lickity-split.
It takes no time at all. Even on a 5000 x 3500 image. :-)

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

Subject: Re: DirectColor on linux
Posted by Vincent Favre-Nicolin on Tue, 23 Nov 1999 08:00:00 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
>
> Vincent Favre-Nicolin (favre@polycnrs-gre.fr) writes:
>
>> Right. My choice, too. Just a change of colors before TV/plot/whatever,
>> for most cases. It's clearly not *hard*, but just annoying since IDL
>> should work for all platforms with very limited changes,and
>> -IMHO-, without any change between one *nix to another.
>
> I'm not above blaming IDL for all the world's problems
> occasionally, but in this case I would have to say it's
> probably not IDL's fault.

Absolutly right, I did not mean to blame IDL. Currently the problem
is due to a limitation of XFree86. I was merely reminding that people
want to use IDL as a platform-independent language. IDL does indeed
quite a good job at being platform-independent.

>> But I have a feeling it will be a bit more
>> funny when using XOR to display text (reversibly)
>> over images. I'm in for some groovy colors =-)

>
> Yep. One reason I *never* use the XOR method. I'm
> a tried-and-true Device Copy man, myself.

Well, I woul like very much not using XOR, but then
I would have to redisplay the image (they can be up to 5000x3500,float)
each time I add some text on it. A bit time and memory-consuming...

Vincent

Subject: Re: DirectColor on linux
Posted by Vincent Favre-Nicolin on Tue, 23 Nov 1999 08:00:00 GMT
View Forum Message <> Reply to Message

Tom Holub wrote:
>
> David Fanning <davidf@dfanning.com> wrote:
>)Vincent FAVRE-NICOLIN (favre@polycnrs-gre.fr) writes:
>)
>)> 3) expensive : buy a commercial X-Windows driver that supports
>)> DirectColor in 24 or 32 bpp.
>)
>)Uh, uh. No. The boss ain't goin' for THAT solution! :-(
>
> Actually my boss would be quite happy with it (And I responded to Vincent
> already in e-mail; I can't imagine why people send e-mail when they've
> also posted a response to Usenet).

Sorry ! I don't spend much time on Usenet unless necessary, so I'm used
to sending also answers via email. Been a bad Useneter :-(, won't do it
again ;-)

> The problem is, it doesn't seem to help.
> Accelerated X claims to support DirectColor, but the behavior
> I saw running in demo mode was the same as under XFree86.

[...]

> That's what I believed, also, but IDL at least lets me set device,direct=24.
> I also tried it while using XIG's Accelerated X server, which does claim
> to support Direct Color, and the behavior, unfortunately, was the same.
> Yes, that's how it looks under XFree86. Under Accelerated X there's
> also a DirectColor entry, but it doesn't seem to have any effect.

IDL will always let you set device, direct=24. It does not mean the
visual is available, until you really use it (use device,/help to test
that).

As for Accelerated X, I do not have any experience. If the available mode for Accelerated X is Directcolor, 24 bit, it is very strange, since this is the default mode for IDL. What are the modes proposed by xdpyinfo ? It may be that the DirectColor mode proposed in AcceleratedX is not 24 bit (?), so that IDL reverts to TrueColor ??
Depending on what xdpyinfo gives, you could use a PseudoColor mode, and still be able to have a private color map.

Vincent

Subject: Re: DirectColor on linux
Posted by davidf on Tue, 23 Nov 1999 08:00:00 GMT
View Forum Message <> Reply to Message

Vincent Favre-Nicolin (favre@polycnrs-gre.fr) writes:

> Right. My choice, too. Just a change of colors before TV/plot/whatever,
> for most cases. It's clearly not *hard*, but just annoying since IDL
> should work for all platforms with very limited changes,and
> -IMHO-, without any change between one *nix to another.

I'm not above blaming IDL for all the world's problems occasionally, but in this case I would have to say it's probably not IDL's fault.

I would say that IDL does (mostly) work the same on all platforms in 24-bit True Color. (I do wish someone would fix the annoying behavior of running 24-bit images through the color tables when Device, Decomposed=0 on Windows platforms, however. That is MOST annoying and doesn't work that way on UNIX machines.) But I don't think there is a general agreement on what Direct Color really means.

I don't know anyone who can get it to work from one machine to another, let alone from one flavor of UNIX to another. And it is not even available in the PC world, so why worry about it. :-)

> But I have a feeling it will be a bit more
> funny when using XOR to display text (reversibly)
> over images. I'm in for some groovy colors =-)

Yep. One reason I *never* use the XOR method. I'm a tried-and-true Device Copy man, myself.

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

Subject: Re: DirectColor on linux
Posted by davidf on Tue, 23 Nov 1999 08:00:00 GMT
View Forum Message <> Reply to Message

Tom Holub (doosh@best.com) writes:

> I can't imagine why people send e-mail when they've
> also posted a response to Usenet).

Uh, well, because sometimes people seem desperate for
an answer and it can (often) take the news a L-O-N-G
time to arrive. :-(

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

Subject: Re: DirectColor on linux
Posted by doosh on Wed, 24 Nov 1999 08:00:00 GMT
View Forum Message <> Reply to Message

In article <383ACAA8.70A7A752@polycnrs-gre.fr>,
Vincent Favre-Nicolin <favre@polycnrs-gre.fr> wrote:
)
)>That's what I believed, also, but IDL at least lets me set device,direct=24.
)>I also tried it while using XIG's Accelerated X server, which does claim
)>to support Direct Color, and the behavior, unfortunately, was the same.
)>Yes, that's how it looks under XFree86. Under Accelerated X there's

)>also a DirectColor entry, but it doesn't seem to have any effect.
)
) IDL will always let you set device, direct=24. It does not mean the
)visual is available, until you really use it (use device,/help to test
)that).

Well, it will let you set it, but then if you try to draw something it
usually will complain with:

% Unsupported X Windows visual (class: DirectColor, depth: 24).
  Substituting default (class: TrueColor, Depth: 24).

) As for Accelerated X, I do not have any experience. If the available
)mode for Accelerated X is Directcolor, 24 bit, it is very strange, since
)this is the default mode for IDL. What are the modes proposed by
)xdpyinfo ? It may be that the DirectColor mode proposed in AcceleratedX
)is not 24 bit (?), so that IDL reverts to TrueColor ??

After looking into this further I have found a way to make it work; in
fact it might have been working before but a silly misconception on my part
kept it from working.

I have a Millennium G200 (8MB) and DirectColor is not listed under my
xdpyinfo under XFree86 3.3.3.1. However, when I run Accelerated X
(http://www.xig.com/), DirectColor *is* listed. I tried the "diddle"
program I posted earlier under Accelerated X a few days ago, and it
didn't seem to work. I just figured out why; the way the window
manager was set up, the diddle window was popping up partially
underneath my GNOME panel; I moved it and then tried to use it, which
didn't work. Since "diddle" is a quick and dirty procedure, it's not
prepared to have its window moved. When I run it without moving the
window, the colors are controlled as expected under DirectColor.

So, it looks like with a fairly low-end (these days) video card and
Accelerated X ($99), it is possible to do DirectColor under
Linux with IDL. Thankfully.
 -Tom

Subject: Re: DirectColor on linux
Posted by J.D. Smith on Wed, 24 Nov 1999 08:00:00 GMT
View Forum Message <> Reply to Message

Vincent FAVRE-NICOLIN wrote:
>
> David Fanning wrote:
>
>> Vincent Favre-Nicolin (favre@polycnrs-gre.fr) writes:

>>
>>> Well, I woul like very much not using XOR, but then
>>> I would have to redisplay the image (they can be up to 5000x3500,float)
>>> each time I add some text on it. A bit time and memory-consuming...
>>
>> Well, not really. With Device Copy you can repair just
>> that portion of the window that you drew something on.
>> That is usually an extremely small region (e.g., a
>> selection box, maybe). This is absolutely lickity-split.
>> It takes no time at all. Even on a 5000 x 3500 image. :-)
>
> The problem is that I am working on X-Ray diffraction images, for which
> I have to index each reflections, ie I must append a small box and three
> integers to 50 to 300 dots on the image. These reflections do cover quite a
> fair amount of the image, so that copying each small zone isn't very
> convenient nor efficient in that case.
> To produce readable text on the image, I have divided the private,
> 256-colors table in two halves, the lower being used to display the
> diffracted intensities (grayscale), and the upper half of the color table is
> a one-color zone, so that text written using XOR appears in that color. That
> works, although I do not consider this very nice code. But I really cannot
> keep a copy of all or parts of the image in this case.
>
> Vincent

You might try keeping a pixmap copy of your image, and just restoring the
whole thing whenever these integer indices change, or quadrants of the whole,
or whatever division is easy and useful. It sounds terribly inefficient, but
depending on the video hardware/software you have, it may not be any worse than
XOR'ing! The reason is that some hardware/software combinations are not able to
write small changes directly to video memory with any efficiency, but instead
buffer and fill large blocks after modification. If you aren't memory limited,
it may work for you. Worth a try anyway. I too used to be a slave to the
perceived efficiencies of XOR, but have since abandoned it after being burned
more than once.

Good Luck,

JD

--
J.D. Smith |*| WORK: (607) 255-5842
Cornell University Dept. of Astronomy |*| (607) 255-6263
304 Space Sciences Bldg. |*| FAX: (607) 255-5875
Ithaca, NY 14853 |*|

Subject: Re: DirectColor on linux
Posted by davidf on Wed, 24 Nov 1999 08:00:00 GMT
View Forum Message <> Reply to Message

David Fanning (davidf@dfanning.com) writes:

> P.S. I'm sending your name to the "IDL Experts"
> nominating committee, but it's hard to say if you
> will hear from them. They haven't had a meeting--
> apparently--in the 12+ years I've been working
> with IDL. At least not one I've been invited to. :-(

My "deep-throat" source just reported in. He (or she) tells me he was in a bathroom stall admiring Ray Sterner's (Expert of the Year, 1993) maps in that new book by William Least Heat Moon, when two members of the Committee walked in discussing this year's secret test question. He put his feet up and they didn't know he (or she) was there.

Anyway, the question is: "How do you get IDL to work on a DirectColor display?" The only hold-up in offering the question to this year's nominees is that they haven't yet been able to find anyone on the nominating committee who knows the answer. :-(

Cheers,

David

P.S. I've heard rumors that there used to be one person who knew, but he left to take a job in the government sector, of all places.

--
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

Subject: Re: DirectColor on linux
Posted by davidf on Wed, 24 Nov 1999 08:00:00 GMT
View Forum Message <> Reply to Message

Vincent FAVRE-NICOLIN (favre@polycnrs-gre.fr) writes:

> To produce readable text on the image, I have divided the private,
> 256-colors table in two halves, the lower being used to display the
> diffracted intensities (grayscale), and the upper half of the color table is
> a one-color zone, so that text written using XOR appears in that color. That
> works, although I do not consider this very nice code. But I really cannot
> keep a copy of all or parts of the image in this case.

Yep. That's what you have to do. I'm convinced you know
what you are doing. I won't offer any more advice. :-)

Cheers,

David

P.S. I'm sending your name to the "IDL Experts"
nominating committee, but it's hard to say if you
will hear from them. They haven't had a meeting--
apparently--in the 12+ years I've been working
with IDL. At least not one I've been invited to. :-(

--
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

Subject: Re: DirectColor on linux
Posted by Vincent Favre-Nicolin on Wed, 24 Nov 1999 08:00:00 GMT
View Forum Message <> Reply to Message

David Fanning wrote:

> Vincent Favre-Nicolin (favre@polycnrs-gre.fr) writes:
>
>> Well, I woul like very much not using XOR, but then
>> I would have to redisplay the image (they can be up to 5000x3500,float)
>> each time I add some text on it. A bit time and memory-consuming...
>
> Well, not really. With Device Copy you can repair just
> that portion of the window that you drew something on.
> That is usually an extremely small region (e.g., a
> selection box, maybe). This is absolutely lickity-split.
> It takes no time at all. Even on a 5000 x 3500 image. :-)

The problem is that I am working on X-Ray diffraction images, for which I have to index each reflections, ie I must append a small box and three integers to 50 to 300 dots on the image. These reflections do cover quite a fair amount of the image, so that copying each small zone isn't very convenient nor efficient in that case.

To produce readable text on the image, I have divided the private, 256-colors table in two halves, the lower being used to display the diffracted intensities (grayscale), and the upper half of the color table is a one-color zone, so that text written using XOR appears in that color. That works, although I do not consider this very nice code. But I really cannot keep a copy of all or parts of the image in this case.

Vincent
