Subject: Re: Baffled by color postscript
Posted by bowman on Mon, 08 Mar 1999 08:00:00 GMT
View Forum Message <> Reply to Message

In article <MPG.114df9eb8bc3875398970e@news.frii.com>, davidf@dfanning.com
(David Fanning) wrote:

> Here is where I am the shakiest. I'm going to assert that
> even if you COULD produce 24-bit PostScript output (and
> I don't think you can from within IDL), there probably
> isn't a printer around that could print it. I say this
> based on my own understanding of printing technology and
> a real quick look around the web for 24-bit color printers.
> The best I found was a printer that claims to print in
> 4-color CMYK color. Anyway, if someone knows better I'd love
> to hear from you.
>
> But that said, I am quite certain that you cannot get 24-bit
> PostScript color out of IDL. (This may not even be an IDL
> problem. I think it likely that the PostScript Level 2
> specification doesn't allow it, although I don't know this
> to be true.)

Hi David,

PostScript definitely supports 24-bit color, and has for years, but the
IDL driver does not - except for bitmap objects in the PS file.

To do a 24-bit bitmap in PS (this is in the documentation) you do

```
SET_PLOT, 'PS'
DEVICE, /COLOR, BITS_PER_PIXEL=8     ;Don't forget /COLOR!
image = BYTARR(8L, 8L, 3L)           ;Make a tiny image
image[*,*,0L] =  8B                  ;Set red plane to 8 (08 in hex)
image[*,*,1L] = 10B                  ;Set green plane to 10 (0A in hex)
image[*,*,2L] = 15B                  ;Set blue plane to 150 (0F in hex)
TV, image, TRUE = 3L                 ;Display image interleaved over dim 3
DEVICE, /CLOSE
```

In the PS file you find:

```
gsave /rstr 8 string def /gstr 8 string def /bstr 8 string def
12700 12700 scale 8 8 8 [8 0 0 8 0 0]
{ currentfile rstr readhexstring pop} bind
{ currentfile gstr readhexstring pop} bind
{ currentfile bstr readhexstring pop} bind
true 3 colorimage
```

```
 08080808080808080a0a0a0a0a0a0a0a0f0f0f0f0f0f0f0f080808080808
08080a0a0a0a0a0a0a0a0f0f0f0f0f0f0f0f08080808080808080a0a0a0a
 0a0a0a0a0f0f0f0f0f0f0f0f08080808080808080a0a0a0a0a0a0a0a0f0f
0f0f0f0f0f0f08080808080808080a0a0a0a0a0a0a0a0f0f0f0f0f0f0f0f
 08080808080808080a0a0a0a0a0a0a0a0f0f0f0f0f0f0f0f080808080808
08080a0a0a0a0a0a0a0a0f0f0f0f0f0f0f0f08080808080808080a0a0a0a
0a0a0a0a0f0f0f0f0f0f0f0f
```

which is an 8x8 24-bit image.

So one ugly alternative for making 24-bit color plots is to plot to a
24-bit X or Z device, TVRD read the image, switch to PS, TV the image, and
then send the PS file to a color printer.

Regards, Ken

--
Kenneth P. Bowman, Professor   409-862-4060
Department of Meteorology   409-862-4466 fax
Texas A&M University   bowmanATcsrp.tamu.edu
College Station, TX   77843-3150   Change the AT to @

Subject: Re: Baffled by color postscript
Posted by bowman on Mon, 08 Mar 1999 08:00:00 GMT

View Forum Message <> Reply to Message

In article <MPG.114df9eb8bc3875398970e@news.frii.com>, davidf@dfanning.com
(David Fanning) wrote:

> Here is where I am the shakiest. I'm going to assert that
> even if you COULD produce 24-bit PostScript output (and
> I don't think you can from within IDL), there probably
> isn't a printer around that could print it. I say this
> based on my own understanding of printing technology and
> a real quick look around the web for 24-bit color printers.
> The best I found was a printer that claims to print in
> 4-color CMYK color. Anyway, if someone knows better I'd love
> to hear from you.
>
> But that said, I am quite certain that you cannot get 24-bit
> PostScript color out of IDL. (This may not even be an IDL
> problem. I think it likely that the PostScript Level 2
> specification doesn't allow it, although I don't know this
> to be true.)

Hi David,

PostScript definitely supports 24-bit color, and has for years, but the
IDL driver does not - except for bitmap objects in the PS file.

To do a 24-bit bitmap in PS (this is in the documentation) you do

```
SET_PLOT, 'PS'
DEVICE, /COLOR, BITS_PER_PIXEL=8     ;Don't forget /COLOR!
image = BYTARR(8L, 8L, 3L)           ;Make a tiny image
image[*,*,0L] =  8B                  ;Set red plane to 8 (08 in hex)
image[*,*,1L] = 10B                  ;Set green plane to 10 (0A in hex)
image[*,*,2L] = 15B                  ;Set blue plane to 150 (0F in hex)
TV, image, TRUE = 3L                 ;Display image interleaved over dim 3
DEVICE, /CLOSE
```

In the PS file you find:

```
gsave /rstr 8 string def /gstr 8 string def /bstr 8 string def
12700 12700 scale 8 8 8 [8 0 0 8 0 0]
{ currentfile rstr readhexstring pop} bind
{ currentfile gstr readhexstring pop} bind
{ currentfile bstr readhexstring pop} bind
true 3 colorimage
 08080808080808080a0a0a0a0a0a0a0a0f0f0f0f0f0f0f0f080808080808
08080a0a0a0a0a0a0a0a0f0f0f0f0f0f0f0f08080808080808080a0a0a0a
 0a0a0a0a0f0f0f0f0f0f0f0f08080808080808080a0a0a0a0a0a0a0a0f0f
0f0f0f0f0f0f08080808080808080a0a0a0a0a0a0a0a0f0f0f0f0f0f0f0f
 08080808080808080a0a0a0a0a0a0a0a0f0f0f0f0f0f0f0f080808080808
08080a0a0a0a0a0a0a0a0f0f0f0f0f0f0f0f08080808080808080a0a0a0a
0a0a0a0a0f0f0f0f0f0f0f0f
```

which is an 8x8 24-bit image.

So one ugly alternative for making 24-bit color plots is to plot to a
24-bit X or Z device, TVRD read the image, switch to PS, TV the image, and
then send the PS file to a color printer.

Regards, Ken

--
Kenneth P. Bowman, Professor   409-862-4060
Department of Meteorology   409-862-4466 fax
Texas A&M University   bowmanATcsrp.tamu.edu
College Station, TX   77843-3150   Change the AT to @

Subject: Re: Baffled by color postscript

Posted by bowman on Mon, 08 Mar 1999 08:00:00 GMT
View Forum Message <> Reply to Message

In article <36E43A2C.176B9C7B@cis.rit.edu>, drwpci@cis.rit.edu wrote:

```
>  OK, I've been through many of David's great web pages, and through the portion
>  of his book related to the subject. Still, I get only monochrome output, or
>  none at all. Here is what I am doing. Note that this works perfectly for the X
>  display (that part between the X-only comments). Running on SGI Octane, IDL
>  5.0 MIPS
>
>  ;;;; X-only code starts
>  ; loop through the data rgb_index() returns the 24 bit color
>  for i = 0, l-1 do begin
>         !p.color = rgb_index(my_rgb[0,i],my_rgb[1,i],my_rgb[2,i])
>         plots, x[i], y[i], psym=4
>  end
```

The IDL PostScript device will display 24-bit color *images*, but will
not display 24-bit line graphics (plot, plots, etc.).

I had the same problem a few months back. Because of the poor
documentation of the PS-device color features, even the RSI support tech
had trouble discovering this.

I filed a request with RSI to add full 24-bit color support to their PS
driver. I hope you will do the same. (And anyone else out there who ever
hopes to do 24-bit color graphics!) I'm disappointed that this has never
been done while scads of other 'useful' features have been added to the
language.

The only workaround I know of is to use a high-resolution 24-bit bitmap
(X-Windows or Z-buffer), write the bitmap to a file, import the bitmap
into a 24-bit-aware program, and then print it. I've been using
GraphicConverter on my Mac, but something like xv would probably work on
Unix.

Ken Bowman

--
Kenneth P. Bowman, Professor   409-862-4060
Department of Meteorology   409-862-4466 fax
Texas A&M University   bowmanATcsrp.tamu.edu
College Station, TX   77843-3150   Change the AT to @

Subject: Re: Baffled by color postscript
Posted by davidf on Mon, 08 Mar 1999 08:00:00 GMT

David R. Wyble (drwpci@cis.rit.edu) writes:

```
>  OK, I've been through many of David's great web pages, and through the portion
>  of his book related to the subject. Still, I get only monochrome output, or
>  none at all. Here is what I am doing. Note that this works perfectly for the X
>  display (that part between the X-only comments). Running on SGI Octane, IDL
>  5.0 MIPS
>
>  ; x,y are 1xn vectors of data
>  ; my_rgb is a 3xn byte values of RGBs for each respective
>  ;    element of x,y
>
>  ; save current device, open postscript
>  thisdevice = !d.name
>  set_plot, 'PS', /copy
>  device, xsize=6, ysize=6, /inches, bits_per_pixel=24, /color
>
>  ; plot the axes
>  plot, x, y, /nodata, xrange=[0,20], yrange=[0,20], title=chartTitle
>  saveColor = !p.color
>
>  ;;;; X-only code starts
>  ; loop through the data rgb_index() returns the 24 bit color
>  for i = 0, l-1 do begin
>   !p.color = rgb_index(my_rgb[0,i],my_rgb[1,i],my_rgb[2,i])
>   plots, x[i], y[i], psym=4
>  end
>
>  ; plot a line at unity, make it the default color
>  !p.color = saveColor
>  oplot, [0,20],[0,20]
>
>  ;;;; X-only code ends
>
>  ; close it up and reset the device
>  device, /close_file
>  set_plot, thisdevice
>
>
>  Any ideas what is wrong? This code produces a postscript file with only the
>  axes and the line at [0,20],[0,20]. From experimenting with SYMSIZE, I believe
>  the points are actually plotting, but they are always white. (When I set
>  SYMSUZE=20, portions of the axes are overwritten, presumably by the large data points.)
```

Well, I'm going to go a little bit out on a limb here,
because the IDL documentation is either silent or (often,
at least to me) ambiguous on the subject.

Here is where I am the shakiest. I'm going to assert that
even if you COULD produce 24-bit PostScript output (and
I don't think you can from within IDL), there probably
isn't a printer around that could print it. I say this
based on my own understanding of printing technology and
a real quick look around the web for 24-bit color printers.
The best I found was a printer that claims to print in
4-color CMYK color. Anyway, if someone knows better I'd love
to hear from you.

But that said, I am quite certain that you cannot get 24-bit
PostScript color out of IDL. (This may not even be an IDL
problem. I think it likely that the PostScript Level 2
specification doesn't allow it, although I don't know this
to be true.)

The fact that you can type these commands is totally misleading:

```
Set_Plot, 'PS'
Device, Bits_per_Pixel=24
```

If you look, the bits per pixel for the device has been set to
8, the largest value allowed:

```
Help, /Device
```

This is output from my Windows NT 24-bit color machine:

```
IDL> set_plot, 'ps'
IDL> device, bits_per_pixel=24
IDL> help, /device
Available Graphics Devices: CGM HP NULL PCL PRINTER PS WIN Z
Current graphics device: PS
 File: <none>
 Mode: Portrait, Non-Encapsulated, EPSI Preview Disabled, Color Disabled
 Offset (X,Y): (1.905,12.7) cm., (0.75,5) in.
 Size (X,Y): (17.78,12.7) cm., (7,5) in.
 Scale Factor: 1
 Font Size: 12
 Font Encoding: AdobeStandard
 Font: Helvetica TrueType Font: <default>
 # bits per image pixel: 8
```

What you CAN do in IDL is write a 24-bit image to a PostScript
file, but you do it in the same way you display a 24-bit image
on an 8-bit display: by writing the 8-bit pixel values to
different channels:

```
   TV, image24, True=1
```

I don't know how this is actually done in PostScript, but the
output certainly looks like what I expect it to look like
when I open the file in a PostScript previewer.

What I have NEVER been able to do is draw a yellow plot on
a charcoal background, by doing something like this:

```
   Plot, Findgen(11), Color='00ffff'xL, Background='707070'xL
```

Even though this command works perfectly well on a 24-bit display.
In fact, I have NEVER been able to see any PostScript output
at all when using this this kind of color specification.

I don't think it is drawing white on white, because I have
tried putting a black image behind the plot and I still don't
see anything. I think, basically, the commands are just
not being written into the file. Or, they are being written
in a way that is not generating an error, but cannot be
interpreted either. I hope someone who knows how this works
can respond.

So, what can you do?

You might try using Color_Quan to convert your RGB color
values into a color table that you could load and use with
the PostScript output. For example, something like this:

```
   IDL> r=[70, 255, 0] & g=[70, 255, 255] & b=[70, 0, 0]
   IDL> val = color_quan(r, g, b, rr, gg, bb)
   IDL> tvlct, rr, gg, bb
   IDL> Plot, Findgen(11), Color=val[1], Background=val[0], /NoData
   IDL> Oplot, Findgen(11), Color=val[2]
```

I think that is about the only thing that has a chance of
working.

Best Regards,

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

[Note: This follow-up was e-mailed to the cited author.]

Subject: Re: Baffled by color postscript
Posted by bowman on Tue, 09 Mar 1999 08:00:00 GMT

View Forum Message <> Reply to Message

In article <36E582B4.E3E8863D@ssec.wisc.edu>, Liam Gumley
<Liam.Gumley@ssec.wisc.edu> wrote:

> David Fanning wrote:
>> Liam Gumley (Liam.Gumley@ssec.wisc.edu) writes:
>>> I guess I'm not sure I understand the problem. I'm able to display true
>>> color images with colored graphics overlays; I've attached a Postscript
>>> example to this message. The postscript file includes 6 test images.
>>> Images 2, 4, and 6 are true color (24 bit) with colored lines (axes,
>>> title, colorbar) overlaid. I've also attached a JPEG version that was
>>> created from the postscript (using ImageMagick).

I think you can use *colors* in the "PLOT" part of the program, but not
256^3 *different* colors (i.e., 24-bit).

Here's an example of a program that works great on a 24-bit X display, but
does not produce the appropriate color PS output. (No bitmaps involved.)

```
PRO TEST24, PRINT = print

IF KEYWORD_SET(print) THEN BEGIN
   SET_PLOT, 'PS'
   DEVICE, /COLOR, BITS_PER_PIXEL = 8, /SCHOOLBOOK, FONT_SIZE=10, /PORTRAIT
   !P.FONT =  0L                              ;Use hardware fonts
ENDIF
PRINT, !P.BACKGROUND

n = 1000L
x = RANDOMU(seed, n)
y = RANDOMU(seed, n)
r = LONG(255*x)
g = LONG(255*y)
b = REPLICATE(0B, n)

PLOT, [0,0], [1,1], /NODATA
PLOTS, x, y, PSYM=1, COLOR = r + 256L*(g + 256L*b)

HELP, /DEVICE
HELP, !P, /STRUCT
```

```
HELP, !D, /STRUCT
TVLCT, r, g, b, /GET
PRINT, r, g, b

IF KEYWORD_SET(print) THEN DEVICE, /CLOSE
SET_PLOT, 'X'
!P.FONT = -1L                          ;Use Hershey fonts

END
```

--
Kenneth P. Bowman, Professor   409-862-4060
Department of Meteorology   409-862-4466 fax
Texas A&M University   bowmanATcsrp.tamu.edu
College Station, TX   77843-3150   Change the AT to @

Subject: Re: Baffled by color postscript
Posted by Liam Gumley on Tue, 09 Mar 1999 08:00:00 GMT

View Forum Message <> Reply to Message

David Fanning wrote:
> Liam Gumley (Liam.Gumley@ssec.wisc.edu) writes:
>> I guess I'm not sure I understand the problem. I'm able to display true
>> color images with colored graphics overlays; I've attached a Postscript
>> example to this message. The postscript file includes 6 test images.
>> Images 2, 4, and 6 are true color (24 bit) with colored lines (axes,
>> title, colorbar) overlaid. I've also attached a JPEG version that was
>> created from the postscript (using ImageMagick).
> Very nice. But can we have a peak at the code that is used
> to generate them. In particular, at the code to generate the
> colored lines. Thanks.

I use a technique I call 'color table splitting'. It involves reserving part of the color table for images, and part of the color table for overlays (e.g. titles, axes, contours etc.).

Try executing the following immediately after starting IDL (requires http://www.dfanning.com/programs/tvimage.pro, and colors.pro attached):

```
;---cut here---
;- Set display mode to true color if available, pseudo color if not

;- Windows case (TRUE_COLOR keyword not supported)

if !d.name eq 'WIN' then $
  device, decomposed=0, retain=2
```

```
;- X and Macintosh case (reverts to pseudo color if true color fails)

if !d.name eq 'X' or !d.name eq 'MAC' then $
  device, true_color=24, decomposed=0, retain=2

;- Create a true color data array

dim = 256
truedata = rebin( indgen( dim ), dim, dim )
data = intarr( dim, dim, 3 )
data[ *, *, 0 ] = truedata
data[ *, *, 1 ] = rotate( truedata, 1 )
data[ *, *, 2 ] = rotate( truedata, 2 )

;- Create byte scaled true color image (reserve colors 0-15)

bottom = 16B
ncolors = !d.table_size-bottom
image = bytscl( data, top=ncolors-1 ) + bottom

;- Load overlay colors from 0 to 15

colors

;- Load greyscale from 16 to !d.table_size-1

loadct, 0, bottom=bottom

;- Display image with color overlay

erase, 7
pos = [0.1,0.1,0.9,0.9]
tvimage, image, /keep, pos=pos
plot, [0], /nodata, /noerase, pos=pos, color=1, title='True Color'
;---cut here---
```

To make this work in Postscript, you need to modify tvimage.pro to accept BOTTOM and NCOLORS keywords thus:

(1) Add the following to the keywords in the PRO statement:

```
BOTTOM=bottom, NCOLORS=ncolors
```

(2) Add the following lines after 'Check for keywords'

```
IF N_ELEMENTS(bottom) EQ 0 THEN bottom = 0
IF N_ELEMENTS(ncolors) EQ 0 THEN ncolors = !d.table_size - bottom
```

(3) Change the line that reads

```
IF true GT 0 THEN LOADCT, 0, /Silent
```

to

```
IF true GT 0 THEN LOADCT, 0, /Silent, BOTTOM=bottom, NCOLORS=ncolors
```

And of course remember to switch to Postscript using

```
SET_PLOT, 'PS'
DEVICE, /COLOR, BITS=8
```

Cheers,
Liam.

Liam E. Gumley
Space Science and Engineering Center, UW-Madison
http://cimss.ssec.wisc.edu/~gumley

```
;---cut here---
PRO COLORS, START = START

;+
; Purpose:
;     Load the sixteen McIDAS graphics colors.
;
; Usage:
;     COLORS
;
; Input:
;     None
;
; Output:
;     None
;
; Optional Keywords:
;     START   Start index in the color table where the graphics
;             colors will be loaded (default = 0).
;
; Notes:
;     The color table assignments are as follows
;      0 => black
;      1 => magenta
;      2 => cyan
;      3 => yellow
;      4 => green
;      5 => red
;      6 => blue
;      7 => white
;      8 => navy
;      9 => gold
;     10 => pink
;     11 => aquamarine
;     12 => orchid
;     13 => gray
;     14 => sky
;     15 => beige
;
; Example:
;colors
;xyouts, 0,   0, 'Magenta', /device, color=1
;xyouts, 0, 100,     'Red', /device, color=5
;xyouts, 0, 200,   'Green', /device, color=4
;xyouts, 0, 300,    'Blue', /device, color=6
;
; Author:
;     Liam.Gumley@ssec.wisc.edu
;-

;- Check keywords

if n_elements( start ) ne 1 then start = 0

;- Load McIDAS graphics colors

r = [0,255,0,255,0,255,0,255,0,255,255,112,219,127,0,255]
g = [0,0,255,255,255,0,0,255,0,187,127,219,112,127,163,171]
b = [0,255,255,0,0,0,255,255,115,0,127,147,219,127,255,127]
tvlct, r, g, b, start

END
;---cut here---
```

Subject: Re: Baffled by color postscript
Posted by davidf on Tue, 09 Mar 1999 08:00:00 GMT

View Forum Message <> Reply to Message

Liam Gumley (Liam.Gumley@ssec.wisc.edu) writes:

> I guess I'm not sure I understand the problem. I'm able to display true
> color images with colored graphics overlays; I've attached a Postscript
> example to this message. The postscript file includes 6 test images.
> Images 2, 4, and 6 are true color (24 bit) with colored lines (axes,
> title, colorbar) overlaid. I've also attached a JPEG version that was
> created from the postscript (using ImageMagick).

Very nice. But can we have a peak at the code that is used
to generate them. In particular, at the code to generate the
colored lines. Thanks.

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: Baffled by color postscript
Posted by davidf on Tue, 09 Mar 1999 08:00:00 GMT

View Forum Message <> Reply to Message

Steve (steves@ruska.nrel.gov) writes:

```
> I don't know if you've seen this, but this is from the manual:
>
> SET_PLOT, 'PS'
>
> ;Set the PostScript device to *8* bits per color, not 24:
>
> DEVICE, FILE='24bit.ps', /COLOR, BITS=8
>
> TV, [[[r]], [[g]], [[b]]], TRUE=3
> DEVICE, /CLOSE
>
> This works very well for me!
```

Well, sure. But the question we are all asking is
why THIS command doesn't work:

```
  Plot, data, Color='00ffff'xL
```

Anyone know the answer to that?

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: Baffled by color postscript
Posted by Steve[2] on Tue, 09 Mar 1999 08:00:00 GMT

View Forum Message <> Reply to Message

David R. Wyble wrote:

> OK, I've been through many of David's great web pages, and through the portion
> of his book related to the subject. Still, I get only monochrome output, or
> none at all.

< snip >

I don't know if you've seen this, but this is from the manual:

```
SET_PLOT, 'PS'

;Set the PostScript device to *8* bits per color, not 24:

DEVICE, FILE='24bit.ps', /COLOR, BITS=8

TV, [[[r]], [[g]], [[b]]], TRUE=3
DEVICE, /CLOSE
```

This works very well for me!

Steve

Subject: Re: Baffled by color postscript
Posted by davidf on Wed, 10 Mar 1999 08:00:00 GMT

View Forum Message <> Reply to Message

Kenneth P. Bowman (bowman@null.tamu) writes:

> I think you can use *colors* in the "PLOT" part of the program, but not
> 256^3 *different* colors (i.e., 24-bit).

Exactly. And the question is "why not?". My usual clandestine
sources are unusually quiet. :-)

But I'm still not sure a printer can PRINT 24-bit color.
It seems to me that even with the top-of-the-line color
printers that there must be a translation between whatever
color system into CMYK. Is it the contention of some of

the PostScript experts here that this is a function of
the printer's RIP?

Perhaps what we should be asking for is not 24-bit
PostScript printing, but something that converts 24-bit
colors into appropriate CMYK colors.

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