Subject: Re: pseudo color and true color
Posted by davidf on Mon, 20 Jan 1997 08:00:00 GMT
View Forum Message <> Reply to Message

Scott Denning <denning@esm.ucsb.edu> writes:

> I am running an idl application under Solaris 2.5 from a Windows NT
> machine using the X/win32 X server.
>
> The Unix box has an 8-bit graphics board. The idl code is built using
> color tables and psedo color through and through. But the NT box has a
> 24-bit video card, and the X server software doesn't allow idl to run in
> Pseudo-Color mode (yes, I've tried setting it with DEVICE).
>
> When I display my plots (color-filled contours and images) on the PC, I
> get indistinguishable shades of purple instead of lovely rainbows
> (colors 0 through 25 in a color space of 16 million).
>
> How can I get around this behavior? Can I somehow generate true color
> images if and only if I'm running my code via the PC X server?

I presume, Scott, that you can't use the Monitors Control Panel (or whatever it's called) on the Windows NT machine to set it up to run in 8-bit mode.

Given that you are stuck with this situation, here is what you can do.

(1) Find out if you are in 24-bit mode. You can use the system variable !D.N_COLORS. If this number is greater than 256, you are running in 24-bit mode.

```
trueColor = (!D.N_COLORS GT 256)
```

(2) You will have to express your contour colors now using an RBG model, rather than the normal color INDEX model. This means that your color value must now be a 24-bit integer that is the equivalent of the color triple associated with the previous color index.

Let me give you an example. Suppose, you have the color yellow. Yellow is represented by the RGB triple (255, 255, 0). In other words, lots of red and green, but no blue.

In an INDEX mode, you would load this color into a color index and use the index to select yellow. For example, if you wanted a yellow plot, you could load this color into color index 10, and use it like this:

```
TVLCT, 255, 255, 0, 10
```

```
Plot, data, Color=10
```

But using the RGB color model, you need a number something like this:

```
number = "bluegreenred"
```

Where "number" is a 24-bit integer, the highest 8 bits representing the blue color, the middle 8-bits the green color, and the lowest eight bits representing the red color you want.

The yellow color you want has the first 16 bits set. That number happens to be 65535. So you could do this:

```
IF trueColor THEN Plot, data, Color=65535L ELSE BEGIN
  TVLCT, 255, 255, 0, 10
  Plot, data, Color=10
ENDIF
```

It is sort of hard to remember the decimal equivalents of color triples (at least for me), so I usually use hexidecimal notation. Then I have two digits of blue info, two of green and two of red. The yellow color can be written like this:

```
Plot, data, Color='00FFFF'x
```

This only confuses me when I want a non-primary color (i.e., most of the time). So what I've done is written a function called COLOR24 that can convert any RGB triple into its equivalent 24-bit integer. You can download COLOR24 from my web page.

So the plot command might look something like this:

```
Plot, data, Color=COLOR24([255,255,0])
```

So, to get back to your contour code, it might have to look more like this:

```
contour_colors = LINDGEN(25)
IF trueColor THEN BEGIN
  TVLCT, r, g, b, /GET
  FOR j=0, 24 DO contour_colors(j) = COLOR24([r(j), g(j), b(j)])
ENDIF
CONTOUR, data, C_COLORS=contour_colors, /FILL
```

(You can modify COLOR24 to process of vector of colors, if you like. I originally designed it for something different.)

(3) If you want to display images in 24-bit mode, you will need a 24-bit image. If you want the image to "look like" it is using the

current color table, you can do something like this:

```
  IF  trueColor THEN BEGIN
     image24 = BYTARR(3, (size(image))(1), (size(image))(2))
     FOR j=0, 2 DO image24(j, *, *) = image
     TV, image24, TRUE=1
     UNDEFINE, image24
  ENDIF ELSE TV, image
```

(The UNDEFINE program is another one you can get from my web page.)

Note, that programs like XLOADCT won't work well in this 24-bit color space. You will have to redisplay any image after you change the colors in order to have the colors on the display reflect the current colors loaded in the "color table".

Sorry for the lengthy explanation. Didn't have time to make it shorter!

Best Wishes and good to hear from you again. How's the new job!

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
