Subject: The Skinny on 24-Bit Color (Long)
Posted by davidf on Tue, 26 May 1998 07:00:00 GMT
View Forum Message <> Reply to Message

Hi Folks,

Thanks to the kind folks at RSI, I have the latest on the
24-bit color situation. Here is what I know about it.

The way 24-bit color is handled changed on BOTH the Windows
and Mac versions of IDL in IDL 5.1. According to RSI, these
platforms have never handled 24-bit color correctly and it
had gotten to the point that people were beginning to count
on the incorrect capabilities in their programming. In the
interest of cross-platform consistency (I'm all for this!)
they decided to fix both platforms to bring them into
compliance with the official 24-bit standards. (Some of this
information, by the way, is spelled out in the IDL 5.1
release notes, but none of it made it into the on-line help
or the official IDL 5.1 documentation.)

Here is the information I found in the release notes:

> Windows:
> In previous releases of IDL on Windows platforms, when
> the system's display properties were set to thousands of
> colors or True Color, decomposed colors were being
> translated by the IDL color table inconsistently.
>
> In IDL 5.1, the Windows behavior with respect to decomposed
> color has been made consistent with other platforms.
> Decomposed colors are now no longer translated by the IDL
> color table; rather, the red, green, and blue components
> of the color are honored directly. To utilize the IDL
> color table, colors must be specified as a single index,
> and decomposed color must be turned off (either by
> setting the system's display properties to 256 colors,
> or by using the following command: DEVICE, DECOMPOSED=0).
>
> Macintosh:
> In previous versions of IDL for the Macintosh the decomposed
> setting on the graphics device with regard to images was the
> opposite of the other platforms. IDL 5.1 now handles the
> decomposed setting properly; DECOMPOSED=0 specifies passing
> indices and DECOMPOSED=1 specifies passing in rgb triplets.
>
> Any programs that depended or attempted to work around the
> aforementioned problem will have to be revised.

I probably don't need to remind people in this newsgroup
that the last sentence certainly means that you will need
to become familiar with the !Version.Release system
variable. :-)

What else does this mean to you if you are a Windows
or Macintosh user?

Well, for starters, it means that if you have decomposed
color turned ON (DEVICE, DECOMPOSED=1), which is often the
default setting on 24-bit devices, then an 8-bit image
will ONLY display in gray scale colors. It means that
the color table tools (XLoadCT or XCOLORS) won't work
at all. It also means that you will have to specify a
24-bit integer to get graphics to display in color:

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

To get an 8-bit image to display in color, or to use a
color table tool, you will have to turn color decomposition
OFF:

```
Device, Decomposed=0
```

Notice that when you do this !D.N_Colors will equal 256,
not the 16.7 million it did in previous versions of IDL.
This, by the way, is how you will know you are on a 24-bit
color device. !D.N_Colors will be greater than or *equal* to
256. (In normal 8-bit Windows or Macintosh versions of IDL
you cannot ever have 256 simultaneous colors.) I don't
know how you will determine unambiguously the depth of the
visual class on an X window device, but more on that later.

If color decomposition is OFF, you will be able to display
color images with the colors loaded in the current color table,
the color table tools will work, and you will be able to
specify a graphics color as an index:

```
TVLCT, 255, 255, 0, 10
Plot, data, Color=10
```

If you change the colors in the current color table, *many*
of us will need to redisplay the image to see the new colors
take effect. What determines whether you are one of the
lucky ones, whose graphical displays change automatically
when you change color tables, is the kind of 24-bit color
device you are using. On Windows, you will be using a True-Color

device. True-Color devices are STATIC color devices. This means that the colors used for the graphics are not tied in any way to the colors loaded in the current color table. If you change the color table values, you MUST redisplay the graphic to see the colors take effect. I presume, but do not know for sure, that the Mac is also a True-Color device.

If you are on a UNIX machine, there is a possibility that you are using a Direct-Color device. (True-Color and Direct-Color visual classes are selected with the DEVICE command.) Direct-Color devices are DYNAMIC color devices. This means that there is a dynamic interaction between the colors of the graphic and the current color table values. Change the values, and you automatically and immediately update the colors in the graphic. There is no need to re-display the image or graphic to see the new colors take effect.

There is currently in IDL no facility to determine if color decomposition is on or off. Nor is there any way to determine which visual class you are currently using or the depth (e.g. 8-bit or 24-bit) of that visual class. According to RSI several programmers have seen a need for such a facility and these requests have been noted. There may be a way to determine such things in future releases of IDL.

I think what this means in the short term is that your programs are going to have to be written so as to protect them from color decomposition being set in such a way as to render them useless. It will be up to each program to determine what kind of color decomposition is needed and to configure the device appropriately. In practice this probably means making sure color decomposition is set correctly *every time you issue a graphics command*. In widget programs this just means that every time you set the current graphics window you also set color decomposition:

```
WSet, (*info).window_index_number
Device, Decomposed = 0
```

Note that you may have to change the color decomposition setting frequently inside a single program. (For example, you may have to display both 24-bit and 8-bit images.)

I still have a number of unanswered questions about color on various devices, but I will be trying to work this out over the next few weeks. It certainly makes

things more complicated as we contend with programs
that must work properly not only on 8-bit and 24-bit
devices, but in different versions of IDL too.
I'll have an article on my web page about this and I
promise to update it frequently as new information
becomes available to me.

As always, I am extremely grateful to get information
from the field. :-)

Best Regards,

David

David Fanning, Ph.D.
Fanning Software Consulting
E-Mail: davidf@dfanning.com
Phone: 970-221-0438
Coyote's Guide to IDL Programming: http://www.dfanning.com/
