
Subject: Re: 8-bit vs. 24-bit color on Windows
Posted by [thompson](#) on Wed, 27 Jan 1999 08:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

davidf@dfanning.com (David Fanning) writes:

> Oh, heck. I'm a sucker for a little newsgroup controversy
> every now and again....

> William Thompson (thompson@orpheus.nascom.nasa.gov) writes:

>> This seems to be a strictly widget-oriented solution. Not everything is
>> widgets!!!! Nor should it be--a lot of specialized data analysis is not done
>> in a widget environment. Quite a bit of it, in fact, is done directly from the
>> command line. Most scientific users who are writing programs their own use
>> don't bother to go to the trouble of writing widget programs.

> While I'm dubious about "most scientific users" working at the command
> line, I'll let that pass. But color table manipulation is NOT
> strictly a widget-oriented solution. It is simply easier to
> implement with widgets. (As are most user-interactive type of
> programs. After all, that is the *point* of a widget program.)

> XCOLORS, for example, works equally well with any object
> that has a "DRAW" method. Give XCOLORS a structure that has
> an object reference, the name of the "DRAW" method to call,
> and the window you want to draw into (even a regular old
> graphics window) and without doing much else you have a window
> that can update itself when the colors in the color table change.

> But if Bill's right, most scientific programmers aren't
> using XCOLORS or XLOADCT anyway. They are building their
> own color table vectors and loading them with TVLCT the
> way God intended. What help can we offer to them?

I didn't say that people weren't using XLOADCT. What I'm saying is that a lot of work (myself included, and the people I know around me) is done using non-widget software. The first thing you do, when you get some data that you don't know what it's going to look like, is you read it and display it. For example, I might say

```
IDL> fxread, 'mydata.fits', array, header ;Generic FITS reader
IDL> tvscl, array
```

Then I want to start playing with the color table, possibly using just LOADCT, possibly with XLOADCT, or possibly with some home-grown equivalent. It's EXTREMELY frustrating if the colors don't change automatically.

To you, maybe that's a minor annoyance which you're willing to put up with for perceived advantages of a more complicated 3-plane color system. For you that's great, and IDL should be able to provide that for you. However, for me, a color-table methodology is much simpler and more appropriate. I argue that IDL should be taking me into account too, and all those more basic users who don't subscribe to this newsgroup.

>> One thing I don't understand is why the graphic needs to be regenerated.
>> Couldn't one just read in the byte values from all active windows and write
>> them back out again?

> Yes. If the byte value represented the color of the pixel,
> which it simply does not in a 24-bit true-color system
> (*any* 24-bit true-color system).

But when I say "tvscf, array", that's what I'm assuming is happening to the image. I'm not working with separate red, green, and blue color values, I'm working with arrays of numbers which I want to appear in a color scheme that can be manipulated to best show what's in the data.

> I don't honestly know. Perhaps RSI *could* hack a PSUEDOCOLOR
> like thing together for the Windows platform. But it sure as
> hell would be non-standard Windows programming and a gigantic
> pain in the neck to maintain, I suspect.

> More to the point, would I want them spending their resources
> doing this, so that we can continue to work with the old
> tired 8-bit strategy far into the future, or do I want them
> adding new features to the language that will allow us to
> write better programs for better and faster computers now?
> Personally, I vote for better programs going forward. And
> which strategy do you think is more likely to result in
> RSI even being around in the future?

Well I don't think this is going to break the back of RSI, and I certainly vote for a pseudo-color capability on Windows just like the other platforms. Almost everybody I know of with a 24-bit graphics cards on a Unix box opts for pseudo-color in IDL. I'm not saying that pseudo-color is more important than 24-bit color, but don't tell me it's less important either!

>> One thing I don't understand is why the graphic needs to be regenerated.
>> Couldn't one just read in the byte values from all active windows and write
>> them back out again?

> Yes. If the byte value represented the color of the pixel,
> which it simply does not in a 24-bit true-color system
> (*any* 24-bit true-color system).

Which is precisely why 24-bit color is more confusing to deal with when you're dealing with what is essentially grey-scale data. It should be simple for RSI to maintain in memory an 8-bit image which then gets translated through the color table into 24-bits when displayed. In pseudo-color mode, any changes to the color table could retrigger this translation automatically, rather than making us have to do it ourselves.

- > I hate to tell you this, Bill. But the problem here is
- > that RSI is hacking operating system code to make it
- > *look* like another operating system, all in the name
- > of cross-platform compatibility. If I recall correctly,
- > that is the solution you are proposing here for yet one
- > more operating system difference. I, for one, could do without
- > any more of these kinds of "solutions". :-)

Without interplatform compatibility, IDL would be a useless product for us. Give that up, and we'd just give up on IDL and use something else instead.

William Thompson
