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:

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

>>  When using IDL under Windows with a 24 bit display setting, the only way
>>  around this problem is to re-display your graphic after changing the
>>  color table. That's why David's XCOLORS program
>>  (http://www.dfanning.com/programs/xcolors.pro) includes a keyword which
>>  enables you to notify an external event handler that the color table has
>>  changed.

> Note that XLOADCT now has a similar capability to call
> an IDL procedure and pass it some "data" when the color
> tables change. (Someone at RSI must *certainly* be
> reading this newsgroup! :-)

(rest deleted)

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 for their own use don't bother to go to the trouble of writing widget programs.

In any case, redisplaying a graphic is a really silly way to go.  Some complicated graphics could take anywhere from seconds to minutes to display--that's a hell of a lot of time and computing power to waste on simply changing a color table.

If the problem on Windows is caused by the Windows environment itself, which I suspect is the case, I still think it should be possible for RSI to emulate the pseudo-color mode we're familiar with.

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?  That wouldn't depend on the window being part of a widget program (or part of a different widget program than the one you told XLOADCT about, for instance), and could be implemented within any color manipulation program without requiring the complication that information about the running programs.

One thing that seems to be happening with IDL lately is that people are being expected to use more complicated programming techniques to solve problems that used to be much simpler.  The problem of displaying traditional pseudo-color

images on the more advanced graphics cards is one such example (although only
on Windows platforms).  Another is the changes within modal widgets, where one
is asked to use complicated things like timer events for things that just used
to work without thinking about it.  It's good that IDL has these more
sophisticated tools for those who can make use of them, but we must remember
that IDL is supposed to be a tool for people who are not professional
programmers.  It's not supposed to be a environment for programmers to write
canned solutions for other people.

William Thompson