
Subject: Re: 8-bit vs. 24-bit color on Windows

Posted by [Martin Schultz](#) on Fri, 29 Jan 1999 08:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

Stein Vidar Hagfors Haugan wrote:

>
> In article <MPG.111a547f221756b5989683@news.frii.com> davidf@dfanning.com
> (David Fanning) writes:
>
>> Liam Gumley (Liam.Gumley@ssec.wisc.edu) writes:
>>>> Almost everybody I know of with a 24-bit graphics cards on a Unix box opts
>>>> for pseudo-color in IDL.
>>>
>>> As I said earlier, this is how everyone here works on Unix boxes.
>> All right. I concede I am a romantic when it comes to software.
>> How else would you dare write a book? :-)
>
> Just to stress the point a bit, I think Bill Thompson is right
> wrt. how many (most?) scientists work with data/color tables,
> at least some of the time.
>
> To them (and me), the "I" in IDL stands for exactly what he
> described (interactive use of commands and xloadct in unison),
> and no widget programmer can ever imagine in advance *everything* a
> scientist would want to do with his data before/during/after
> displaying them (the scientist doesn't know in advance, either:-),
> so the interactive command line is an extremely valuable part of IDL.
>
> Not having the possibility of exploring an image by tweaking
> the color table in pseudo-color mode would take away a lot of the
> original appeal of IDL, IMHO, so this should be taken very seriously
> by RSI.
>

So true, so true !

> I think a large market segment would feel left out if the color-table
> methodology got lost...
>

Didn't someone say lately that RSI sells (was it ?) 60% as PC/Windows versions these days? Don't let me repeat this GENPLOT experience again. This was once a very nice program that just never made it to windows (not to speak of Unix), because the programmer held a flag for IBM in the OS/2 vs. Win3.1 war. In some ways, I still mourn over that loss, because it was even simpler to do this command line data exploration with GENPLOT than it is with IDL (it's all these little inconsistencies that make IDL such a "human" language ;-). Now, I really don't want to

loose IDL to the clicking community (programs that run automatically without requiring any user interaction do have their virtues!) and this old batch processing principle just runs counter to all widgeting (and maybe even objects). It's great to have both options in IDL, just as it is great to have X windows that allow you to type commands instead of DOS boxes that you have to close when they are done. Unfortunately, it's us scientists who have the greatest demands in terms of what a program should be able to do (i.e. graphics, numerical algorithms) and this in a very incoherent fashion. And, it is usually us scientists who can't afford spending too much on software (how many people are still using 4.x versions of IDL?). So, I just hope that neither IDL for Unix nor the colortable scheme will ever be abandoned!!

A little besides the point, but since this thread has broadened anyhow

...

Let me unravel a little dream here that I had again lately: Wouldn't it be great if a program would have enough intelligence to notice inconsistencies in user input? Example: you draw a map and you specify both hatched and filled continents. Instead of somehow selecting either one, a window would pop up and present you with the possible options. Likewise, all parameters and keywords from a program could be displayed (and queried) automatically either by setting a special keyword (e.g. draw,/query) or as some kind of error handler (e.g. if (n_params() eq 0) then query,...). This could be helpful to run programs that you are not too familiar with (now, what were these keywords again ??), and it would take care of those countless situations when you run a program in different contexts or with different data sets where for instance one variable is missing that was explicitly defined in the other data set (or it is named differently). The mechanism would still be "invisible" if you specify all parameters and keywords correctly, and it would of course have to "know" the keywords and parameters automatically so that you don't have to pass a long name list to query when you call it. The "interactive variable assigner" should allow to either enter values manually, or select from all variables that are "visible" on the caller level (and maybe even globally)...

OK. Gotta get back to work now,
Martin.

--

Dr. Martin Schultz
Department for Engineering&Applied Sciences, Harvard University
109 Pierce Hall, 29 Oxford St., Cambridge, MA-02138, USA

phone: (617)-496-8318

fax : (617)-495-4551

e-mail: mgs@io.harvard.edu

Internet-homepage: <http://www-as.harvard.edu/people/staff/mgs/>
