Subject: Re: Understanding Color in IDL
Posted by JD Smith on Wed, 11 Oct 2006 21:14:46 GMT
View Forum Message <> Reply to Message

On Wed, 11 Oct 2006 13:24:55 -0600, David Fanning wrote:

> JD Smith writes:
>
>> Every time I interact with new IDL users, even very savvy, technically
>> inclined users, the number one initial stumbling block relates to
>> colors, usually DECOMPOSED issues. I think ITTVIS should take a hard
>> look at their default color model, and ask themselves whether it's
>> satisfying the largest number of users. Yes, once you learn the magic
>> tricks, you are fine, but first impressions are everything, and the
>> first impression many of my colleagues have of IDL is that it is buggy
>> and can't display images very well.
>
> I guess we could argue about what would "satisfy the
> largest number of users", but I have a feeling the iTools
> crowd (and I'm not saying that tongue in cheek) and even
> the folks like me, who LOVE 24-bit color and couldn't
> live without it now that I've built enough tools to deal
> with it, would get the shaft. In my experience, 80-90%
> of the IDL users still use the indexed color model for
> most of their programming needs.

And why wouldn't they? Decomposed TrueColor is useful when you have a specific set of colors in mind, approximating, as you mention, how a human would see it with their own eyeballs (for instance as digital cameras attempt to do). Most data which flows into IDL isn't obtained with devices which attempt any such "as it would appear in the real-world" approximation, but rather instruments whose data requires some form of visual representation to mesh with the evolutionarily-encoded image analysis skills of their human operators. Indexed color tables are the fastest route to that sort of visualization.

The fact that color tables can only be 256 elements long in IDL is a vestige of 8b-bit video cards, which had their own 256 element hardware color index table built right into card, which instantly changed all colors on the screen when modified (leading to the "color-flashing" most of us try not to remember). This vestigial length limit is, however, a separate issue from whether color tables are useful at all. I for one would be glad to use 16-bit color tables on devices which could support them. Aside from the larger number of colors available, what other advantages do you see of the decomposed model?

For me, a real revolution would be to allow separate, independent, and fully private color tables, so you wouldn't have to constantly worry that someone else stomped on your color space just because they needed to use a color or two. The ability to:

get_color_table_slot,slot,/FREE
loadct,0,SLOT=slot

and re-use that slot (perhaps sharing it among a few apps), never having to modify it again (unless you wanted to), would be really helpful. Give us about 2^16 slots, and that should suffice.

> I'd be satisfied with the way it is now (since we finally
> have most of the UNIX folks able to see *something* other
> than red on black plots!) if we just didn't have to write
> our programs differently to send then to a PostScript file
> or printer. :-(

Fully agree on that point.

JD
