
Subject: Re: Understanding Color in IDL
Posted by [JD Smith](#) on Thu, 12 Oct 2006 01:05:59 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Wed, 11 Oct 2006 15:40:41 -0600, David Fanning wrote:

> JD Smith writes:
>
>> 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.
>
> I have no problem with color tables. Couldn't live without
> them in every scientific program I ever wrote. My beef is
> with a dumb TV command that can't figure out for itself
> whether I have a 2D array that should go through a color
> table or a 24-bit image that has its color information built
> in and should NEVER go through a color table.

No argument there ;).

> I like decomposed color because I want to use that expensive
> graphics card I bought and I want my image displayed with one
> color table and I want other colors used for my beautiful
> graphics display, and I want it all at once. I don't want
> to have to piggy back on my image colors or (worse) sacrifice
> image colors for drawing colors.

Indexed color on TrueColor devices *is* using that fancy graphics card. In fact, there was a brief period when some people really preferred 8bit cards, since "fiddling" the color map to bring out detail in an image (an activity which is thankfully waning in fashionability) operated very quickly, because it just involved writing 256 values to that on-board color table hardware, whereas for 24bit cards (which were just arriving at the time) you had to reencode and rewrite the *entire* image for each "fiddle", over and over. Now cards are fast enough that this objection no longer exists, but you could argue that using color tables on 24bit cards can be very taxing, with all the continuous redraws.

If we could keep multiple separate color tables trivially, I think

you'd like color tables just as much. In fact, you could do this now, but it would require some color overseer object to keep track of and edit the entire (single) colormap. I think IDL should do this for us, at a much lower level, with multiple colormaps, deferring the decision of which to apply until the point in the code where the data is actually run through the lookup table. Somehow it doesn't seem right to preface every plotting/drawing command with a complete reset of the internal color table. In that context, decomposed color is a winner.

JD
