Subject: Re: A defense of decomposed color
Posted by JD Smith on Mon, 06 Nov 2006 17:26:43 GMT
View Forum Message <> Reply to Message

Thanks for all the thoughts, David. I hadn't appreciated the subtle differences between postscript and printer devices w.r.t. color tables. I suppose I am struggling with the concept because of the typical "free-form" design of applications I work on. They might enter into a small function to draw a specific overlay polygon many different ways, and this function might be entered 100's of times for a given notional redraw (tied to a motion event, even). A single 5us penalty compared to drawing a large image is utterly negligible. However, I just tested a simple plots call, and wrapping it in "device,decomposed=1,get_decomposed=gd & ... & device,decomposed=gd" increases the time required by roughly a factor of 5. If you need to perform say ten thousand of these for a given screen redraw (even if sent to a double buffer), it would be the difference between updates occuring 100 times/s or 20 times/s (which would be noticeable). Now, in reality, your budget may well be dominated by other overheads in all sensible cases, but it just doesn't feel right to toggle the drawing mode back and forth many thousands of times per second.

Though color tables are stone-age tools, now I can simply:

a) Set up a color table with N image colors and N_max-N "system" or plotting colors.

b) Reload that color table on enter events.

c) Plot and/or display whenever, wherever, without having to futz the color model beforehand.

Obviously, this requires a device,decomposed=0 in the startup file (which is simply documented), but it saves considerable time and trouble. The drawback is, of course, different tools have different conventions, and you always end up with conflicts.

So it appears there is no ideal solution. It also appears the IDL color model is somewhat broken. Ideally, there would be no decomposed state, and IDL primitives would "do the right thing" with the data given them. Is this how Matlab works?

JD