## Subject: Re: Understanding Color in IDL Posted by Foldy Lajos on Thu, 12 Oct 2006 09:09:42 GMT

View Forum Message <> Reply to Message

On Wed, 11 Oct 2006, David Fanning wrote:

```
> Folks,
>
> A quick look at my web page statistics for the keywords people
> use to find my pages convinces me that understanding how color
> works in IDL is still the number one problem for new users.
> (Closely followed by how to get decent PostScript output and
> how to calculate the log base 2 of a number, apparently.)
>
> I've written a number of articles on color over the years,
> but I've never collected this material into a single
> coherent explanation. I've attempted to do that now with
> a chapter I've written for the 3rd edition of my book.
> (Please don't hold your breath, but I *am* working on it,
> off and on.)
>
> I've made the chapter available as a PDF file for
> those who might be interested. If you are still confused
> after reading it, please let me know. I'd like to get
> this right before the book is published. :-)
>
  You can find a link to the PDF file here:
   http://www.dfanning.com/color_tips/colorchp3.html
>
>
>
  Cheers.
 David
> P.S. Actually, the number one way people come to my web page
> is searching the Google image archives for "pretty girl oktoberfest".
 but I don't want to get into the reasons behind that!
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming: http://www.dfanning.com/
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")
>
```

Hi David,

what I miss is a summary table for quick reference. Here is what I use for

FL, on all platforms and for all devices (X, WIN, PS, PDF, Z):

 $C = color (0...2^24-1)$  $c = C \mod 256 (0...255)$ 

r = red component of C  $(0...255) = C \mod 256$ 

q =green component of C  $(0...255) = C / 256 \mod 256$ 

b = blue component of C (0...255) = C / 256 / 256

R = lookup table for red (256 element vector)

G = lookup table for green (256 element vector)

B = lookup table for blue (256 element vector)

R,G,B for R.G.B for decomposed=1 decomposed=0

24 bit image R[r], G[g], B[b] r, g, b

R[c], G[c], B[c] 8 bit image C, C, C

R[c], G[c], B[c] lines, text, etc. R[r], G[g], B[b]

It slightly differs from IDL behaviour. Also note the R[r], G[g], B[b] values, which are decomposed and indexed at the same time (these values differ from r, g, b only if a non-linear lookup table is used).

regards, lajos

ps: probably I will change the 'lines with decomposed=1' mode to omit table lookup, to be compatible with IDL.