
Subject: Re: image color representing a vector...

Posted by [Peter Mason](#) on Wed, 16 Jun 1999 07:00:00 GMT

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

John Stanly Videen <user@internet.com> wrote:

```
<...>
> for x=0,127 do begin
>     for y=0,127 do begin
>
>         red = ei(x,y) * ev(x,y,0)    ; x direction
>         green = ei(x,y) * ev(x,y,1)   ; y
>         blue = ei(x,y) * ev(x,y,2)    ; z
>
>         out(x,y) = new_function(red, green, blue)
>
>     endfor
> endfor
>
> tv, out
<...>
```

Somewhat hesitantly, from what you've got here I'd say that you're just looking for a function to take "unscaled" values "red", "green" and "blue" and get a coloured pixel from them.

If by some chance you don't have a copy of David Fanning's book on IDL programming techniques then I'd shamelessly recommend that you get a copy. He covers this stuff well.

But anyway, here are some ideas...

First, do the calc in one hit using array arithmetic:

```
rgb_unscaled = REBIN(REFORM(ei,nx,ny,1),nx,ny,3) * ev
(Rgb_unscaled has dimensions [nx,ny,3] and is probably float or double?)
```

Next, scale to byte. This could be as simple as:

```
rgb_scaled=BYTSCL(rgb_unscaled)
or it might be done band by band with selected minima and maxima, and
perhaps with each band going through some non-linear stretch (like
histogram normalisation), etc.
```

Finally, if you have a hi- or truecolor display you can display this result immediately with: `TV,rgb_scaled,TRUE=3`

If you have an 8-bit display then you must first construct a 1-byte paletted image and corresponding colour table with `COLOR_QUAN()`, e.g., `rgb_paletted=COLOR_QUAN(rgb_scaled,3,rlut,glut,blut,COLORS=256)`. This can then be displayed with `TV,rgb_paletted & TVLCT,rlut,glut,blut`.

I hope this is of some help.

Peter Mason

Sent via Deja.com <http://www.deja.com/>
Share what you know. Learn what you don't.
