

Brian Larsen writes:

> I am not sure I agree with you entirely on this, if it were as
> hopeless as you say then why do we make those plots, our eyes can
> obviously decode the "meaning" in the color. Since you have the
> colorbar, it seems that gives a one-to-one mapping of colors to data
> value. At the worst case maybe just at the points in the colorbar
> where you have the value.
>
> If I ask the question another way does it make more sense? How can
> one choose out of an image a particular color and then display that?
> For example in that aerosol image of yours say I wanted to know where
> the data was between 3 and 3.5 (red), I can presumably "select" red
> and then pull out the red pixels from the image? No? Is this as easy
> as the 3 elements in `image[* ,0, 0]` and a `where` statement? I am not
> too clear on exactly how the colors are built.

Normally, the color encoding of the color bar is different from the color encoding of a 24-bit image. The former typically uses indexed color, the latter used decomposed color. They *may* be related. For example, suppose we load a "yellow" color in color index 10:

```
r = 255 & g = 255 & b = 0  
TVLCT, r, g, b, 10
```

And we decide that all pixels in our data between 3 and 3.5 are going to be colored with yellow:

```
index = Where(data ge 3 and data le 3.5)
```

We color our 24 bit image with this color:

```
s = Size(data, /Dimensions)  
image = BytArr(s[0], [s[1])  
rimage = image [index] = r  
gimage = image [index] = g  
bimage = image [index] = b  
image24 = [[[rimage]], [[gimage]], [[bimage]]]
```

Now, the question is, how can you go from the 24-bit image back to the data value? We can click on the image. We learn that the pixel we clicked on has a value (255, 255, 0). Can we find such a color triple in the color table? Maybe, we have

to search for the proper value in each of the three color vectors at the same time. We find that color located in bin 10 of the color table vectors. What does that tell us about the data values? Absolutely nothing. I could have chosen to represent data between 2000 and 3000 with this color index.

To make sense of this, you will have to know what color index 10 represents in the color table. You may have this information somewhere. But you **don't** have it in the image. You can't go anywhere with an index value of 10 and recover useful information from either your images or your data array.

Cheers,

David

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.")
