Subject: Re: color_quan & gif files
Posted by davidf on Fri, 30 Jul 1999 07:00:00 GMT
View Forum Message <> Reply to Message

Simon Hall (Simon.Hall@atm.ch.cam.ac.uk) writes:

>> A couple of quick questions before I make a stab at an
>> answer:
>>
>> 1. Are you certain you are in 24-bit color and not
>> 16-bit color?
>
> Definitely 24 bit.
>
>> 2. Why do you think the white background is gray? :-)
>> I guess I am asking, where are you viewing these GIF
>> images? If you read them into IDL after writing them,
>> do they look OK? What about displaying them in Photoshop?
>> If you are viewing them in a browser, which one, specifically?
>
> They look grey in netscape 4.5, IE5 and if I load them into
> PaintShop Pro the background is [248,248,248]

Well, here is what I think is happening and how I fixed
it in my example program.

I think it has to do with how the color space is partitioned.
There must be several ways to do this and IDL provides two
different methods. A statistical method, using a variation
of the Median Cut Algorithm, is the default and usually
produces better colors when there are a lot of colors in
the image. But I think the Floyd-Steinburg dithering method
gives more accurate colors, when the image has just a few
colors. This method is selected by using the CUBE keyword
in the Color_Quan function.

I found this code produced white whites (255, 255, 255) for me. :-)

Here is my example image:

```
window, xsize=200, ysize=200
device, decomposed=10
polyfill, [0, 0.5, 0.5, 0, 0], [0, 0, 0.5, 0.5, 0], $
  /Normal, Color=getColor("red", 1)
polyfill, [1.0, 0.5, 0.5, 1.0, 1.0], [0, 0, 0.5, 0.5, 0], $
  /Normal, Color=getColor("green", 2)
polyfill, [1.0, 0.5, 0.5, 1.0, 1.0], [0, 0, 0.5, 0.5, 0], $
  /Normal, Color=getColor("blue", 3)
```

```
  polyfill, [0, 0.5, 0.5, 0, 0], [1.0, 1.0, 0.5, 0.5, 1.0], $
    /Normal, Color=getColor("white", 4)
```

And here is the code to produce the GIF file:

```
  image3d = TVRD(True=1)
  image2d = Color_Quan(image3d, 1, r, g, b, Cube=6)
  Write_GIF, 'test.gif', image2d, r, g, b
```

A further advantage of the CUBE method is that the colors it produces are independent of the input image. Thus, it is likely to work better with primary colors of the sort used in plots. Rather than, say, the colors used in images.

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438 E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155
