Subject: Re: COLOR_QUAN question
Posted by davidf on Thu, 19 Aug 1999 07:00:00 GMT
View Forum Message <> Reply to Message

Struan Gray (struan.gray@sljus.lu.se) writes:

> A while back someone asked for a way to force COLOR_QUAN to use a
> particular color table. I tried seeding it with a dummy image
> containing 256 pixels and a simple ramp through the desired table,
> hoping to use /MAP_ALL and /GET_TRANSLATION to force subsequent calls
> to use the same colour map.
>
> I gave up because the colour map returned by the first call was
> always different from that used to generate the dummy image. That is,
> even if you feed COLOR_QUAN an RGB image with only 256 pixels (which
> it should be able to reproduce exactly with an 8-bit colour table) the
> RGB values of the colour map it generates are different from the
> pixels' RGB values.

I just tried a similar experiment. The image has 256 values
(a 3D image constructed of a 2D image created like this:
image2D = BytScl(Findgen(256)#Findgen(256))).

Oddly enough, the resulting 2D image has only 155
values. And even though the returned color tables
differ in the pixel values above 154, they appear
identical in those values from 0-154. The
images *appear* identical on the display, though,
of course, they can't be.

But, again, I'm not sure this is a big deal. Color_Quan
is used to produce images that *LOOK* the same. That is
to say, it is a *display* routine, not a *processing*
routine. Performing image processing steps on the
resulting image, or using the pixel values to mean
something, is just as big a mistake as using the
byte values of any image on the display, rather than
the original image data.

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/