
Subject: Re: Color Tables with Gif files

Posted by [gurman](#) on Tue, 10 Jan 1995 03:19:25 GMT

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

In article <3es7nf\$pkp@post.gsfc.nasa.gov>, fisher@echo.gsfc.nasa.gov
(Brad L. Fisher) wrote:

> I am trying to generate an animation sequence of radar images in Gif format.
> I am using IDL to modify the images by placing the input gif image into a
> larger window, annotating it with time, date, place information and adding
> a color bar appropriately scaled. The problem I am having is that the r,g,b
> color table for each image is scaled differently . Thus, a color bar
> generated for any given image can only be applied to that image, even though
> the actual colors represented in the image correspond to the same numerical
> value from one image to the next. So for example if a value of 40 dBZ is
> assigned to the color red, this will be true for all images generated. The
> problem comes about because of the way the data values are scaled within gif.
> My question then is is there a way of generating a color bar that will remain
> fixed from one image to the next while each image is allowed to vary depending
> on the scaling. It seems that this is not possible since you can only
load one
> color table at a time and that is the color table you are stuck with. Thus,
> the color bar would have to change along with the image, which in my case can
> not be done in automated fashion, at least not that I am aware of. I hope
> that I have not answered my own question. If anyone has any suggestions about
> how this problem can be addressed I would greatly appreciate it.
>

Brad -

We came up against much the same problem in creating QuickTime movies of solar soft X-ray images. We make the movies on a Macintosh, but start with PICT files generated on an Alpha in IDL. I search through all the images for min and max once to come up with a common scaling, then iterate one more time to apply that scaling to each original image. Time consuming (well, not all that time consuming on an Alpha) for 500 - 600 images, 512 x 512 pixels each, but it works.

When in doubt, use brute force, I guess. The alternative would require 600 Mbyte of memory, which IDL doesn't seem happy about accessing for a single datacube (although one of our Alphas does have 320 Mbyte, nothing we have is quite up to putting the whole movie in an array in memory.... yet. Unfortunately, the movie, composed of once-a-day images, keeps growing). I wish IDL worked better with virtual memory; I've never understood why it doesn't.

Joe Gurman

--

Joseph B. Gurman / NASA Goddard Space Flight Center/ Solar Data Analysis Center / Code 682
/ Greenbelt MD 20771 USA / gurman@uvsp.gsfc.nasa.gov

| Federal employees are still prohibited from holding opinions while at work. Any opinions
expressed herein must therefore be someone else's. |
