Subject: Re: View 24-bit images
Posted by Liam E. Gumley on Fri, 04 Feb 2000 08:00:00 GMT
View Forum Message <> Reply to Message

"Alvaro T." wrote:

> Thanks to both. Now I understand it. But I still have the same problem
> because I only have a matrix of, for example, 100x100 values with data
> from -32000 to 32000 (is that Long, Int...? I don't know...). Should I
> change the type data before draw it?
> Imagine that a guy send to you a 100x100 matrix with data from -32000 to
> 32000 values. How will you show it in the screen? I don't need show it in
> colors, only in grey levels.

Alvaro,

I think we're getting to the root of the problem now. When you said '24-bit' images in your original post, I assumed you meant a True Color image with red, green, and blue components. However I think you actually have a short or long integer array that you wish to display as a Pseudo Color image. Here are some tips:

(1) To find the type and size of the array, use 'help', e.g.

```
image = dist(256)
help, image
```

(2) When reading from HDF files, IDL preserves the data type. So if the array was a short integer in the HDF file, it will be a short integer in IDL.

(3) Regardless of the type of your array (short, long, float, double), it must be scaled to byte values in order to be displayed as an image. If you wish to use built-in IDL commands for byte scaling, you can use either TVSCL or BYTSCL, e.g.

```
tvscl, image
tv, bytscl(image)
```

Both of these commands display a scaled version of your image, where the minimum to maximum array values are linearly mapped to byte image values in the range 0 to !D.TABLE_SIZE - 1. If you wish to select a certain data range in the input array, then BYTSCL accepts the MIN and MAX keywords, e.g.

```
tv, bytscl(image, min=0.0, max=125.0)
```

(4) IMDISP byte-scales the input array by default, either from minimum to maximum, or over a range you specify, e.g.

```
imdisp, image
imdisp, image, range=[0.0, 125.0]
```

IMDISP also offers a plethora of other advantages which you can read about at
http://cimss.ssec.wisc.edu/~gumley/imdisp.html

Cheers,
Liam.
http://cimss.ssec.wisc.edu/~gumley