Subject: 16 bit / 8 bit depth colors on the mac
Posted by Rick Shafer on Sat, 02 Nov 1996 08:00:00 GMT
View Forum Message <> Reply to Message

Someone *please* tell me how to get genuine 16 bit colors on my mac with IDL. The default basicly packs the three RGB values into 15 bits, cutting off the lowest 3 significant bits, which is NOT what I want.

I used to know the magic incantation to do this, but have forgotten.

Please, reply by e-mail... (and yes I tried to look for this in the FAQ, not to mention RTFMing, toggling various key word values, etc. to no avail.) (I won't even talk about how IDL crashes if I should change the screen depth while running...

Subject: Re: 16 bit / 8 bit depth colors on the mac
Posted by davidf on Tue, 05 Nov 1996 08:00:00 GMT
View Forum Message <> Reply to Message

Rick Shafer <rick.shafer@gsfc.nasa.gov>writes:

> Someone *please* tell me how to get genuine 16 bit colors on my mac with
> IDL. The default basicly packs the three RGB values into 15 bits,
> cutting off the lowest 3 significant bits, which is NOT what I want.
>
> I used to know the magic incantation to do this, but have forgotten.

I think your question is not how to get thousands of colors on your Macintosh, because the answer is to set your monitor to thousands of colors and use 3D images. For example:

```
TV, image3d, TRUE=1
```

I am sure you know this. Rather, I suspect your question has to do with your data. You probably have 16-bit unsigned data and you want to know how to display that image data properly in IDL, which does not have an unsigned integer data type. That is a different question.

Here is what you can do. Read the image data into 16-bit integers (the default or short IDL integer size). To convert the signed integer array to an unsigned integer array, you will have to convert the array to long integers, like this:

```
image = INTARR(256, 256)
READU, lun, image
image = LONG(image) AND 'FFFF'x
```

(I learned this in a news post compliments of Bill Thompson at NASA
Goddard and Peter Mason at CSIRO in Australia.)

Now, I know on *my* Mac I would want to display this kind of
data in 256-color mode, so that color tables, etc. still work.
I set my monitor to 256-color mode and display the image
like this:

```
   TV, BYTSCL(image, TOP=!D.N_COLORS-1)
```

> (I won't even talk about how IDL crashes if I should change the
> screen depth while running...

Uh, IDL!? I don't think so. Try MacOS. (This from a devoted Mac user.)

David

--
David Fanning, Ph.D.
Phone: 970-221-0438
Fax: 970-221-4728
E-Mail: davidf@fortnet.org

Subject: Re: 16 bit / 8 bit depth colors on the mac
Posted by davidf on Thu, 07 Nov 1996 08:00:00 GMT
View Forum Message <> Reply to Message

Peter Mason writes:

```
>  It just occurred to me that there is a way to view an unsigned int (16-bit)
>  image without having to convert to LONGs.   (Memory may sometimes be an
>  issue, especially for large multiband images.)
>  e.g.,
>      image = INTARR(256, 256)
>      READU, lun, image          ;read in the unsigned int image
>      f=fix(32768)               ;F is a signed short int, value = -32768
>      image=temporary(image)+f    ;remap "unsigned" values to monotonically
>                                 ;increasing signed values
>      tvscl,image
>
>  The problem with viewing unsigned int data as if they are signed is that
>  values 32768 .. 65535 get interpreted (backwards!) as -32768 .. -1.
>  (Values 0 .. 32767 are ok.)
>
>  By subtracting 32768 from the data we're mapping to an acceptable signed
>  int range:
>    0     .. 32767       =>      -32768 .. -1
```

> 32768 .. 65535 => 0 .. 32767
> So any operation which is concerned with the RELATIVE data range (like
> TVSCL or BYTSCL) stands a chance of working on the remapped data.

Umm, perhaps. The problem with this Peter, at least as I see it, is that there are too many people already who think when they view an image that the are "seeing their data". They are not. They are viewing an *abstraction* of their data. Namely, their data displayed in the number of colors available on their display device.

This will encourage people to work with that abstraction as if it were the real thing. I'm always leary of this, because I think it leads naturally to bad decisions about what the data means. I would rather take the extra hit on bytes for the long integers and know what my "real" data is.

David

****
* David Fanning, Ph.D.
* 2642 Bradbury Court, Fort Collins, CO 80521
* Phone: 970-221-0438 Fax: 970-221-4762
* E-Mail: davidf@dfanning.com
*
* Sometimes I go about pitying myself, and all along my
* soul is being blown by great winds across the sky.
* -- Ojibway saying
****