
Subject: Re: accessing large arrays quickly
Posted by [davidf](#) on Thu, 27 May 1999 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

David Mattes (dmattes@u.washington.edu) writes:

> what would be nice though, is a routine similar to tvscl, but with all the
> flexibility of bytscl, and with a couple extra features. tvscl only
> allows specifying a top value. it would be nice to specify a bottom
> value, to make working with split color tables a lot easier. does anybody
> know of such a procedure???

Well, because you asked and because I didn't feel like doing anything else today, here is a TVSCALE program that has the flexibility of BYTSCL and a "couple of extra features".

You can use it like this:

```
TVScale, image, TOP=200, BOTTOM=100
```

and the image data will be scaled between 100 and 200. Note that this is 101 pixel values, however. So your color table should be loaded like this:

```
LoadCT, 5, NColors=101, Bottom=100
```

As for additional features. Here are a few:

1. It will erase the display if required.

```
TVScale, image, /Erase
```

2. The image can be positioned in the window with the POSITION keyword.

```
TVScale, image, Position=[0.1, 0.2, 0.9, 0.75]
```

3. The program is device independent. In other words, it works exactly the same way in the PostScript device as it does on the display. You don't have to size images two different ways depending upon which device you are in.

4. It works transparently with 8-bit or 24-bit images. You don't have to worry about setting the appropriate TRUE keyword, nor do you have to worry about whether you are using DECOMPOSED color or not. The program will set these keywords appropriately for the image. (Well, it will if you are using IDL 5.2, which has the ability to get and set these keywords.)

5. It will work with !P.MULTI.

```
!P.Multi = [0, 2, 1]  
Plot, Histogram(image)  
TVScale, image, /Multi
```

6. By default it fills up the entire current graphics window.

7. If you prefer, you can choose to keep and preserve the aspect ratio of the image:

```
TVScale, image, /Keep_Aspect
```

You can find the program here:

<http://www.dfanning.com/programs/tvscale.pro>

As some of you will realize, it's companion program is TVImage, the one I prefer:

<http://www.dfanning.com/programs/tvimage.pro>

> also, if anyone can answer this question, you get serious kudos!

Oh, I *like* serious kudos! :-)

```
> i am  
> defining several objects and structures in my programs. if i add a tag to  
> a structure once this structure has existed in memory, i must restart IDL  
> and recompile, in order for my changes to be accepted. i always perform  
> garbage collection on all the heap variables, using heap_gc. how can i  
> get IDL to forget about all previous structure definitions???
```

Exit IDL!!! (This was too easy.)

Uh, seriously. This is how this is *suppose* to work. If you are changing your structure definitions all the time, you should be using anonymous rather than named structures. Can you imagine the confusion that might ensue if IDL allowed you to change COMMON block or named structure definitions whenever you "felt like it"?

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
