
Subject: Colormaps (a favorite subject!)

Posted by [Craig Hamilton](#) on Fri, 24 Mar 2000 08:00:00 GMT

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

Hi all:

I've run into a little snag dealing with colormaps and thought I would seek the advice of the wise ones.

I have an image-viewing program that displays gray-level images. Works great. I had this idea that I wanted to make some of settings user-selectable on startup, so I first popped up a window with some radio buttons in it. One set of radio buttons lets the user select the number of gray-levels used in the colormap.

The problem is that IDL requires something like:

```
window,0,colors=numcolors,/pixmap,xsize=10,ysize=10  
wdelete,0
```

to initially setup the colormap, prior to any windows being created.

But, I have already created my popup window. So it seems to me I'm in a catch-22: I have to specify the number of colors before I create my popup window, but I don't know the number of desired colors until after I have created my popup window.

My popup doesn't create any draw widgets, if that matters.

So, the question is:

Is it possible to use a popup window to get from the user the number of desired gray-levels?

I hope this is clear. Any tips most appreciated.

Craig

Subject: Re: Colormaps (a favorite subject!)

Posted by [davidf](#) on Tue, 28 Mar 2000 08:00:00 GMT

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

Craig Hamilton (cah@medeng.wfubmc.edu) writes:

> I must not be explaining myself very well.

No, we have all just gotten used to people explaining themselves so poorly that we refuse to take anything at face value anymore. I'm sure it is an occupational hazard on our part. :-)

I understood your point, I just assumed the problem lay elsewhere. My apologies.

No, I don't think it is possible to do what you want, except in the ways that have been pointed out to you, which all suffer from inelegance in one way or the other. I think probably a start-up file is the way to go. At least the user has some control about it. Although the downside is that he or she has to know what they are doing ahead of time. A sorry state of affairs, usually.

> PS: I really appreciate all the time you spend helping us
> IDL novices out.... where do we mail the beer?

Just put it in with your book order. :-)

Cheers,

David

P.S. PV-Wave used to be capable of changing the number of colors used in the current session as long as there were no graphics windows open. This was a VERY useful innovation that IDL has never seemed to pick up on. But it would have sure solved your problem.

--

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

Subject: Re: Colormaps (a favorite subject!)
Posted by [Liam E. Gumley](#) on Tue, 28 Mar 2000 08:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Kenneth P. Bowman <kbowman@null.net> wrote in message
news:kbowman-2803001837260001@tl6-218-030.tca.net...
> In article <38E0C93C.AF2BD78D@ssec.wisc.edu>, "Liam E.Gumley"
> <Liam.Gumley@ssec.wisc.edu> wrote:
>
>> I would also
>> encourage you to start thinking about designing you application to work

>> in 24-bit graphics mode, where color flashing problems do not exist, and
>> you always have 256 color table entries available.
>
> And where you cannot do 24-bit PostScript output to print your pretty
> 24-bit displays.
>
> Seriously, I agree with Liam that an all 24-bit world nice. I'm very
> frustrated by the inability to make hard copies of 24-bit graphics other
> than through bitmaps. This is becoming more and more of a problem as our
> displays change to 24-bit and my graphics tools (want to) change with
> them.

Ken,

It seems like the PRINTER device allows decomposed 24-bit colors to be used.
The following example produces a blue on white plot in IDL 5.3 on my
Windows98 PC. When dialog_printersetup is called, I select the 'HP Color
Laserjet PS' printer, with output directed to a file.

```
result = dialog_printersetup()  
set_plot, 'PRINTER'  
device, /true_color  
plot, indgen(10), color='FF0000'XL  
device, /close
```

Something similar could be probably be configured in IDL for UNIX. Let me
know how it goes.

Cheers,
Liam.

Subject: Re: Colormaps (a favorite subject!)
Posted by [kbowman](#) on Tue, 28 Mar 2000 08:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

In article <38E0C93C.AF2BD78D@ssec.wisc.edu>, "Liam E.Gumley"
<Liam.Gumley@ssec.wisc.edu> wrote:

> I would also
> encourage you to start thinking about designing you application to work
> in 24-bit graphics mode, where color flashing problems do not exist, and
> you always have 256 color table entries available.

And where you cannot do 24-bit PostScript output to print your pretty
24-bit displays.

Seriously, I agree with Liam that an all 24-bit world nice. I'm very

frustrated by the inability to make hard copies of 24-bit graphics other than through bitmaps. This is becoming more and more of a problem as our displays change to 24-bit and my graphics tools (want to) change with them.

Ken

Subject: Re: Colormaps (a favorite subject!)

Posted by [John-David T. Smith](#) on Tue, 28 Mar 2000 08:00:00 GMT

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

Alex Schuster wrote:

>

> Craig Hamilton wrote:

>

>> Thanks for the code, but my problem is not with grey-levels vs.

>> number of colormap entries. My problem is as originally stated

>> it: I want the user to be able to select the number of colors

>> in the colormap, but I can't popup a window to get that information

>> without having the number of colormap entries already set

>> in the act of popping up the window. I must not be explaining

>> myself very well. I've done loads of image processing programming

>> in other languages on Unix and Windoze, so I'm familiar

>> with colormaps and scaling of image data for display.

>> I'm working with 16-bit data on 8-bit displays right now and I want to

>> avoid color-flashing, so I want to use from 32 to 128 colormap

>> entries for this display program. I want the user to be

>> able to select how many entries are used.

>

> Why don't you just let IDL allocate as much colors as it can get (window

> & wdelete & ncolors=!d.n_colors)?

>

>> Maybe a little more simply:

>> I have been initializing the size of the colormap used with:

>> window,0,colors=numcolors,/pixmap,xsize=10,ysize=10

>> wdelete,0

>> But now I just want to get the number 'numcolors' from the user first.

>> I think I am realizing that I cannot get that number with a widget.

>> Right?

>

> I have had a similar problem: A tool displays brain images, but there

> are often too few colors. So I check for the number of available colors,

> and if it is at least 64, I use 64 color cells. If there are less, I

> install a private colormap. This results in a flickering display, but at

> least the user gets enough colors.

>

> What I did to check the number of available colors is to open a

- > temporary IDL session which checks the number, and writes it to a file
- > in /tmp. The session then exists, and another IDL session starts. It
- > reads the file in /tmp, and allocates as much color cells as stated in
- > the file.

I think Craig is looking for a way to let the user control how many colors IDL is stealing from the colormap. You can certainly configure this in many ways, but I think he wants more control than letting it steal all remaining colors, or all but some specific number of remaining colors (using negative numbers in COLOR=numcolors). What if he wanted two versions of IDL running at once (though see the SET_TRANSACTION keyword to DEVICE in this case), or wanted to later start another display tool? If IDL's color greed is not put in check, there will be none remaining for these other applications.

Unless this is something which users will want to change all of the time, I would let them configure it once, write it to file, and instruct them that they must restart IDL for changes to be in effect (Windows users will not be disconcerted by this requirement whatsoever). Then, upon startup, read in the file, and set that number of colors. They can always change it again with the stipulation that it will go into effect on the next session only.

JD

--

J.D. Smith |*| WORK: (607) 255-5842
Cornell University Dept. of Astronomy |*| (607) 255-6263
304 Space Sciences Bldg. |*| FAX: (607) 255-5875
Ithaca, NY 14853 |*|

Subject: Re: Colormaps (a favorite subject!)

Posted by [Alex Schuster](#) on Tue, 28 Mar 2000 08:00:00 GMT

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

Craig Hamilton wrote:

- > Thanks for the code, but my problem is not with grey-levels vs.
- > number of colormap entries. My problem is as originally stated
- > it: I want the user to be able to select the number of colors
- > in the colormap, but I can't popup a window to get that information
- > without having the number of colormap entries already set
- > in the act of popping up the window. I must not be explaining
- > myself very well. I've done loads of image processing programming
- > in other languages on Unix and Windoze, so I'm familiar
- > with colormaps and scaling of image data for display.
- > I'm working with 16-bit data on 8-bit displays right now and I want to
- > avoid color-flashing, so I want to use from 32 to 128 colormap
- > entries for this display program. I want the user to be

> able to select how many entries are used.

Why don't you just let IDL allocate as much colors as it can get (window & wdelete & ncolors=!d.n_colors)?

> Maybe a little more simply:

> I have been initializing the size of the colormap used with:

> window,0,colors=numcolors,/pixmap,xsize=10,ysize=10

> wdelete,0

> But now I just want to get the number 'numcolors' from the user first.

> I think I am realizing that I cannot get that number with a widget.

> Right?

I have had a similar problem: A tool displays brain images, but there are often too few colors. So I check for the number of available colors, and if it is at least 64, I use 64 color cells. If there are less, I install a private colormap. This results in a flickering display, but at least the user gets enough colors.

What I did to check the number of available colors is to open a temporary IDL session which checks the number, and writes it to a file in /tmp. The session then exists, and another IDL session starts. It reads the file in /tmp, and allocates as much color cells as stated in the file.

Alex

--

Alex Schuster Wonko@weird.cologne.de
alex@pet.mpin-koeln.mpg.de

PGP Key available

Subject: Re: Colormaps (a favorite subject!)

Posted by [Liam E. Gumley](#) on Tue, 28 Mar 2000 08:00:00 GMT

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

Craig Hamilton wrote:

> David:

> Thanks for the code, but my problem is not with grey-levels vs.

> number of colormap entries. My problem is as originally stated

> it: I want the user to be able to select the number of colors

> in the colormap, but I can't popup a window to get that information

> without having the number of colormap entries already set

> in the act of popping up the window. I must not be explaining

> myself very well. I've done loads of image processing programming

> in other languages on Unix and Windoze, so I'm familiar

> with colormaps and scaling of image data for display.

> I'm working with 16-bit data on 8-bit displays right now and I want to

> avoid color-flashing, so I want to use from 32 to 128 colormap

> entries for this display program. I want the user to be
> able to select how many entries are used.
>
> Maybe a little more simply:
> I have been initializing the size of the colormap used with:
> window,0,colors=numcolors,/pixmap,xsize=10,ysize=10
> wdelete,0
> But now I just want to get the number 'numcolors' from the user first.
> I think I am realizing that I cannot get that number with a widget.
> Right?

Craig,

Here's an alternative approach.

First, let IDL decide how many colors are available. You can do this via a startup file. I'll assume that you wish to support 8-bit graphics only. Here is the startup file I would use.

```
if !version.os_family eq 'unix' then device, pseudo=8
window, /free, /pixmap, colors=-10
wdelete, !d.window
device, decomposed=0, retain=2, set_character_size=[10, 12]
device, get_visual_depth=depth
print, 'Display depth: ', depth
print, 'Color table size: ', !d.table_size
```

This causes a graphics window to be opened when IDL starts up, thus allowing IDL to determine the size of the color table. This approach is more flexible than selecting a pre-set number of colors.

Then in your application, let the user choose which part of the color table will be used. This is typically done via the BOTTOM and NCOLORS keywords. BOTTOM refers to the bottom entry in the color table, and NCOLORS refers to the number of entries in the color table that should be used. The default values would be

```
bottom = 0
ncolors = !d.table_size - bottom
```

To scale and display an image:

```
image = dist(256)
loadct, 13, bottom=bottom, ncolors=ncolors
tv, bytscl(image, top=(ncolors - 1) + byte(bottom))
```

Using this method, you can display multiple images with different color tables. For example:

```
image = rebin(dist(32), 256, 256, /sample)
window, /free
bottom = 0
ncolors = 64
loadct, 13, bottom=bottom, ncolors=ncolors
tv, bytscl(image, top=(ncolors - 1)) + byte(bottom)
```

```
image = dist(256)
window, /free
bottom = 64
ncolors = 64
loadct, 3, bottom=bottom, ncolors=ncolors
tv, bytscl(image, top=(ncolors - 1)) + byte(bottom)
```

This technique is called 'color table splitting', and it can be very useful when IDL is running in 8-bit graphics mode. I would also encourage you to start thinking about designing your application to work in 24-bit graphics mode, where color flashing problems do not exist, and you always have 256 color table entries available.

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

Subject: Re: Colormaps (a favorite subject!)
Posted by [Phillip David](#) on Tue, 06 Jun 2000 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Craig;

I've been away from this newsgroup for a LONG time, so I'm sorry that my answer is 3 months out of date. I'm sure you've solved the problem by now, one way or another. However, one possibility that wasn't mentioned is to provide your users with a command-line option on startup to allow them to set this value. It's not as convenient to change as a pop-up (when you don't want the default), but it's actually less intrusive for those who do want the default.

If it's an issue of certain users wanting it, they probably won't mind having to say something like `your_app, ncol=64`, especially if they can set up a script to do this the way they want, and that's a one-time thing.

Phillip

Craig Hamilton wrote:

>
> Hi all:
> I've run into a little snag dealing with colormaps and thought I would seek
> the advice of the wise ones.
>
> I have an image-viewing program that displays gray-level images.
> Works great. I had this idea that I wanted to make some
> of settings user-selectable on startup, so I first popped up a window with
> some radio buttons in it. One set of radio buttons lets the user select
> the number of gray-levels used in the colormap.
>
> The problem is that IDL requires something like:
> window,0,colors=numcolors,/pixmap,xsize=10,ysize=10
> wdelete,0
>
> to initially setup the colormap, prior to any windows being created.
>
> But, I have already created my popup window. So it seems to me
> I'm in a catch-22: I have to specify the number of colors before I
> create my popup window, but I don't know the number of desired
> colors until after I have created my popup window.
>
> My popup doesn't create any draw widgets, if that matters.
>
> So, the question is:
> Is it possible to use a popup window to get from the user the number
> of desired gray-levels?
>
> I hope this is clear. Any tips most appreciated.
>
> Craig
