Subject: Re: zimage question.....(David, are you out there?)
Posted by davidf on Sat, 09 Oct 1999 07:00:00 GMT
View Forum Message <> Reply to Message

Richard G. French (rfrench@wellesley.edu) writes:

> This is really a question for David F. but since it involves color, I
> thought others might like to try it out and tell me what I am doing
> wrong.

> I'm using David's ZIMAGE routine - a nice image zoomer - and trying to
> add a few bells and whistles - that all seemed to work fine - but here
> is what I don't understand.
>
> zimage, image
> ; displays a nice image, I can zoom, change colors, etc.
>
> zimage, anotherimage
> ; displays a nice image, which I can also zoom, etc.
>
> Both windows stay on the screen at the same time. Here is the part I
> don't understand. (I'm running on 24-bit color screen)
>
> Let's say in the first instance of zimage, I say that I want to change
> the colors. The color table widget shows up, and I can change the colors
> in the first instance, along with the zoom window associated with this
> instance.
>
> Then, I go to the second instance, and say that I want to change the
> colors of THAT set of windows. If I have not killed the previous
> color table widget, that is the one the gets reactivated, and the colors
> don't get changed on the second set of windows, as I would like, but on
> the first set of windows. Somehow, the event handler is not figuring out
> which instance of zimage is doing the calling, and the second set of
> windows does not get updated.
>
> I've tried various permutations of this, such as closing the color
> widget
> and trying to start it up from scratch in the second instance of zimage,
> but I cannot find a reliable way to get it to set the colors on the
> windows
> from which I called the color table widget.

My only excuse is that ZIMAGE was written a LONG time ago,
well before I started answering 10-15 questions a day about
how colors work in IDL. :-)

In fact, the XCOLORS program that ZIMAGE uses to change color

tables has long been capable of having more than one of itself
on the display at the same time. (Note that this is NOT possible
with XLOADCT, since XLOADCT uses common blocks and must protect
itself appropriately.) But because I don't want an *unlimited*
number of XCOLORS programs on the display, I decided that you
could have as many as you like as long as each instance
had a unique name. This way, I can easily have a color table
tool for each resizeable graphics window I create. (This was
the original requirement for XCOLORS, in fact.)

The simplest way to create an XCOLORS tool with an unique
name is to create the title of the program with the
window index number of the graphics window XCOLORS should
change colors for. I do it like this (here is the code as I
modified it in the ZIMAGE program, which you will find below):

```
   XColors, Group=event.top, NColors = info.ncolors, $
      Bottom=info.bottom, NotifyID=[event.id, event.top], $
      Title='ZImage Colors (' + StrTrim(info.drawIndex,2) + ')'
```

The only other trick is that the programs colors should
be loaded *before* XCOLORS is called, because XCOLORS
always starts up with the colors in the *current* color
table. In ZIMAGE, this meant that I needed to add the
R, G, and B vectors of the color table to the info
structure. Not a bad idea, in any case, because this
also gives me the opportunity to protect the colors
in the window from other programs or users changing
the color table.

As long as I was mucking about, I also decided to add
a NoInterpolation keyword, which would insure that nearest
neighbor sampling instead of bilinear interpolation is
used for the zoomed image.

You can find the newly modified ZIMAGE program in the
usual place:

   http://www.dfanning.com/programs/zimage.pro

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/