
Subject: 24-bit/8-bit color (Re: read_tiff - simple question)

Posted by [steinhh](#) on Mon, 01 Mar 1999 08:00:00 GMT

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

David Fanning wrote:

> P.S. I've just received the latest batch of IDL books from
> the printer. This version has been updated rather extensively
> for IDL 5.2. I have also increased the amount of discussion
> on 8-bit verses 24-bit image display, as this continues to
> be a source of confusion for many IDL users as we make the
> transition from 8-bit to 24-bit computers.

So what's the "upgrade fee" :-)

Seriously, I'd like to take this opportunity to add something to the discussion we had a while ago, concerning pseudo-8-bit behaviour etc:

I (and some others) expressed concern that RSI should be doing something about the inability (on 24-bit PC displays) to do e.g.

TVSCL,IMAGE & XLOADCT

from the command line in order to "explore" an image by tweaking the color map. I believe I (we) speak for a substantial user group in this matter - having to redisplay an image to see color table changes is just **not** acceptable for command line work.

Why should users of "new and improved" hardware + "new and improved" IDL versions have to go through a lot of programming hassle (or use canned programs) in order to do something that could be done in an intuitive way in the "good old days"...

So, along the same lines that IDL does offer graphics window backing store, I think RSI should offer the possibility of "value" backing store, i.e. storing the 8-bit color table index and automatically updating the display colors whenever a new color table is loaded.

Having thought about this for a while, I think it would be best if this was done on a "per-window" basis. I.e., when creating a new window, one could specify whether or not it was intended for interactive (command line) color table manipulation. That way, the full 24-bit color scale would be available to [widget] programs, while at the same time allowing the user to do explorative color table changes in non-program windows (this is all the more important now that we have the active command line alongside

widget programs).

It would also be nice to make the DECOMPOSED=0/1 status be a property of whatever window is currently set, to avoid having to set this status in programs every time graphics output is about to occur.

I realize that there will be some penalty wrt. efficiency, but this penalty will be a lot smaller if the functionality is offered by IDL internally, rather than by **always** requiring the use of programmed "TV wrappers" with callback functions to deal with both decomposition and color table updates.

As an extra bonus, it would be **very** nice if one could specify more than one such "interactive colormap" - i.e. along the lines of:

```
window,0      ;; Default window creation, means "use the
               ;; current colormap (let's call it "map1"),
               ;; in interactive mode ("value" backing store)

tvsc1,image1   ;; Display image1 with this color map

window,1      ;; Create new window, still using map1

tvsc1,image2

xloadct       ;; Manipulate current map, with instant
               ;; update of window 0 and 1 (since they're
               ;; both using the map that's manipulated).

newmap = obj_new('pseudomap') ;; Create a new (interactive)
                               ;; colormap
window,2,pseudo=newmap ;; Both windows share this new map
window,3,pseudo=newmap ;;
:
xloadct ;; Manipulates current (i.e. "newmap") color table,
        ;; updating window 2 & 3, no effect on 0 and 1.

wset,0 ;; Now we're switching to a window using map1
xloadct ;; Manipulate map1 again.. no effect on window
        ;; 2 and 3, but instant update of 0 and 1.
```

In the case of an 8-bit display, all the color tables would probably be identical, with changes to one reflected in all windows. Or maybe one could still have different color tables internally, and the effective one (the one that's actually communicated to the screen hardware) would depend on which window

was selected/active (wset)?

Any comments/improvements?

Regards,

Stein Vidar

Subject: Re: 24-bit/8-bit color (Re: read_tiff - simple question)

Posted by [David Foster](#) on Tue, 09 Mar 1999 08:00:00 GMT

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

Stein Vidar Hagfors Haugan wrote:

>

> David Fanning wrote:

>

>> P.S. I've just received the latest batch of IDL books from
>> the printer. This version has been updated rather extensively
>> for IDL 5.2. I have also increased the amount of discussion
>> on 8-bit verses 24-bit image display, as this continues to
>> be a source of confusion for many IDL users as we make the
>> transition from 8-bit to 24-bit computers.

>

> So what's the "upgrade fee" :-)

>

> Seriously, I'd like to take this opportunity to add something to
> the discussion we had a while ago, concerning pseudo-8-bit
> behaviour etc:

>

> I (and some others) expressed concern that RSI should be doing
> something about the inability (on 24-bit PC displays) to do e.g.

>

> TVSCL,IMAGE & XLOADCT

>

> from the command line in order to "explore" an image by tweaking
> the color map. I believe I (we) speak for a substantial user
> group in this matter - having to redisplay an image to see color
> table changes is just *not* acceptable for command line work.

>

> Why should users of "new and improved" hardware + "new and
> improved" IDL versions have to go through a lot of programming
> hassle (or use canned programs) in order to do something that
> could be done in an intuitive way in the "good old days"...

>

> So, along the same lines that IDL does offer graphics window
> backing store, I think RSI should offer the possibility of
> "value" backing store, i.e. storing the 8-bit color table index

> and automatically updating the display colors whenever a new
> color table is loaded.
>
> Having thought about this for a while, I think it would be best
> if this was done on a "per-window" basis. I.e., when creating a
> new window, one could specify whether or not it was intended for
> interactive (command line) color table manipulation. That way, the
> full 24-bit color scale would be available to [widget] programs,
> while at the same time allowing the user to do explorative color
> table changes in non-program windows (this is all the more
> important now that we have the active command line alongside
> widget programs).
>
> It would also be nice to make the DECOMPOSED=0/1 status be a
> property of whatever window is currently set, to avoid having to
> set this status in programs every time graphics output is about
> to occur.
>
> I realize that there will be some penalty wrt. efficiency, but
> this penalty will be a lot smaller if the functionality is
> offered by IDL internally, rather than by *always* requiring the
> use of programmed "TV wrappers" with callback functions to
> deal with both decomposition and color table updates.
>
> As an extra bonus, it would be *very* nice if one could specify
> more than one such "interactive colormap" - i.e. along the lines
> of:
>
> window,0 ;; Default window creation, means "use the
> ;; current colormap (let's call it "map1"),
> ;; in interactive mode ("value" backing store)
>
> tvsc1,image1 ;; Display image1 with this color map
>
> window,1 ;; Create new window, still using map1
>
> tvsc1,image2
>
> xloadct ;; Manipulate current map, with instant
> ;; update of window 0 and 1 (since they're
> ;; both using the map that's manipulated).
>
> newmap = obj_new('pseudomap') ;; Create a new (interactive)
> ;; colormap
> window,2,pseudo=newmap ;; Both windows share this new map
> window,3,pseudo=newmap ;;
> :
> xloadct ;; Manipulates current (i.e. "newmap") color table,

> ;; updating window 2 & 3, no effect on 0 and 1.
>
> wset,0 ;; Now we're switching to a window using map1
> xloadct ;; Manipulate map1 again.. no effect on window
> ;; 2 and 3, but instant update of 0 and 1.
>
> In the case of an 8-bit display, all the color tables would
> probably be identical, with changes to one reflected in all
> windows. Or maybe one could still have different color tables
> internally, and the effective one (the one that's actually
> communicated to the screen hardware) would depend on which window
> was selected/active (wset)?
>
> Any comments/improvements?
>
> Regards,
>
> Stein Vidar

One word...ditto!

In our research we constantly require very interactive changes of gray-scale color-tables, and 24-bit color simply does not do what we need. All of our major widget programs require 8-bit color (so this is important for more than just command-line work).

I sincerely hope that Research Systems does not underestimate the number of users that will be very upset if they drop support for 8-bit color in any way, shape or form.

I like the suggestions for supporting interactive 24-bit color, and especially the suggestion for window-specific color-tables. I haven't thought enough about this to hazard a guess as to the feasibility of these suggestions, or the performance hit resulting from their implementation, but if they would work, I would consider them to be a VERY valuable feature.

Just my .03 (inflation).

Dave Foster

--

~~~~~  
David S. Foster      Univ. of California, San Diego  
Programmer/Analyst   Brain Image Analysis Laboratory  
foster@bial1.ucsd.edu   Department of Psychiatry  
(619) 622-5892      8950 Via La Jolla Drive, Suite 2240  
La Jolla, CA 92037

~~~~~