Subject: Re: changing contrast and brightness on the fly
Posted by John-David T. Smith on Tue, 19 Jun 2001 23:18:52 GMT
View Forum Message <> Reply to Message

Simon Williams wrote:
>
> Greetings,
>
> I'm looking for tips on how to implement an image display feature that's
> bugging me. I'm new to widget programming, trying to get up to speed
> with David Fanning's book and other helps, but any short-cuts would be
> appreciated.
>
> The job is to display an MRI image and to be able to adjust the image
> brightness and contrast interactively, without going to any special
> widgets like sliders. The control I have in mind is to
> middle-click the image and then have drag right/left control contrast
> and drag up/down control brightness.
>
> The plan is to replicate functionality that the end users (radiology
> folks) are already familiar with from other image viewers. In the
> longer term plan, the display would also be re-sizable and allow
> interactive ROI drawing as well.
>
> It sounds easy but I can't find anything similar described on the web
> etc. to use as a suitable starting point. Question: is changing the
> contrast and brightness to be achieved by re-defining the
> color table and re-scaling the data?

A standard feature of astronomical viewers. For a solution which uses only colormap fiddling (vs. full image rescaling), see atv:

http://cfa-www.harvard.edu/~abarth/atv/atv.html

This brings up the age old question of how best to dynamically redisplay images (brightnest/contrast/etc.). With only 255 colors, making the top 100 white to increase brightness is pretty wasteful, and cuts down on the dynamic visual range... much better (and slower) is to rescale the image range of interest into the full colormap.

Here's how Andrew (and cohorts) did it:

```
++++++++++++++++++++++++++++++++++++++++++++
pro atv_stretchct, brightness, contrast,  getmouse = getmouse

; routine to change color stretch for given values of
; brightness and contrast.
; Complete rewrite 2000-Sep-21 - Doug Finkbeiner
```

```
; This routine is now shorter and easier to understand.

common atv_state
common atv_color

; if GETMOUSE then assume mouse positoin passed; otherwise ignore
; inputs

if (keyword_set(getmouse)) then begin
   state.brightness = brightness/float(state.draw_window_size[0])
   state.contrast = contrast/float(state.draw_window_size[1])
endif

x = state.brightness*(state.ncolors-1)
y = state.contrast*(state.ncolors-1) > 2   ; Minor change by AJB
high = x+y & low = x-y
diff = (high-low) > 1

slope = float(state.ncolors-1)/diff ;Scale to range of 0 : nc-1
intercept = -slope*low
p = long(findgen(state.ncolors)*slope+intercept) ;subscripts to select
tvlct, r_vector[p], g_vector[p], b_vector[p], 8

end
++++++++++++++++++++++++++++++++++++++++++++++
```

I think it's more intuitive to offset from the current location at
button press (as opposed to absolute position in the window as ATV seems
to do). If you want to spend some time and do it the gold label way,
you could record all mouse positions (x_i,y_i) since button press, and
determine a "speed" of motion from the separation of subsequent
positions (e.g. x=1,2,3,4 vs. x=1,5,9,14), increasing the rate of change
of colormap stretch/mouse movement for faster speeds (a method Apple
pioneered for mouse movement in the MacOS). Tuning this for platform
independence might be difficult, however.

JD

P.S. I also just saw ATV uses the hidden widget_text keyboard event
hack I discovered it seems so long ago... The world is small and funny.

P.P.S. Common blocks bad. Bad common block, bad.

P.P.P.S. My object-oriented direct graphics viewer with plug-in support
and the message passing paradigm is being facelifted for heavy duty work
with a satellite instrument analysis package next year. When it gets in
shape, I'll release it. Some interesting plug-in's already written:
Box statistics, central aperture photometry, image slicing/plotting, box

histogram scaling, color-map selection, brightness/contrast selection, special purpose data interfaces (plane selection, project-specific data browsing), etc. It's very malleable.