Subject: Re: Colour maps overlaid on grey-scale (medical) images
Posted by davidf on Thu, 11 Mar 1999 08:00:00 GMT
View Forum Message <> Reply to Message

Jason Brookes (jason.brookes@rmsb.u-bordeaux2.fr) writes:

> I would like to know how to display colour overlays on medical images.
> For example, an overlay of bloodflow rate superimposed (in hot body
> colour scale) on grey-scale images of the brain. At the moment, I am not
> able to do this without obliterating the information in the original
> image. Is it possible to overlay a colour map onto a grey-scale image
> without obliterating information in the grey-scale image ? ie: by making
> the colour overlay "transparent" to some degree ?

There are probably more sophisticated ways to do this
(and I would like to hear about them), but here is a
quick and dirty method that has always worked quite
well for me.

The idea is to "half-tone" your image so that each
adjacent pixel is from the other image. By creating
two color tables and scaling the original images
appropriately into them, you can get a resulting image
that looks pretty darn close to what you want.

Here is a little example program using the elevation.dat
and ctscan.dat data sets in the IDL distribution. You
can download the LoadData program from my web page:

  http://www.dfanning.com/programs/loaddata.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/
Toll-Free IDL Book Orders: 1-888-461-0155
--

```
Pro Color_On_Gray, image_1, image_2

  ; Get the data sets if needed.

IF N_Params() EQ 0 THEN BEGIN
```

```
   image_1 = Loaddata(7)
   image_2 = Loaddata(5)
ENDIF

   ; Size the second image to fit the first.

s = Size(image_1, /Dimensions)
image_2 = Congrid(image_2, s[0], s[1], /Interp)

   ; Load the color tables. Gray-scale and Red Temperature

ncolors = !D.Table_Size
halfcolors = Byte(ncolors / 2)
LoadCT, 0, NColors=halfcolors
LoadCT, 3, NColors=halfcolors, Bottom=halfcolors

   ; Scale the data. First image uses gray-scale.

image_1 = Bytscl(image_1, Top=halfcolors-1)
image_2 = Bytscl(image_2, Top=halfcolors-1) + halfcolors

   ; Create a vector for pixelation.

x = Findgen(s[0]/2) * 2

   ; Pixelate the image.

image = BytArr(s[0], s[1])
image[x, *]   = image_1[x, *]
image[x+1, *] = image_2[x+1, *]
image[*, x]   = Shift(image[*, x], 1)

   ;  Display image.

Window, XSize=s[0], YSize=s[1], /Free
TV, image

END
```