Subject: Re: Overlaying where data
Posted by pgrigis on Fri, 25 Jan 2008 20:53:28 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
> jtmcahill@gmail.com writes:
>
>> Ok, let's see if I can explain this more clearly. First, I display
>> the original image in tvscl.
>
> If you are really using TVSCL to display your image, I think
> you are already in trouble or you will be in trouble soon.
> If colors matter to you, forget you ever heard anything about
> TVSCL. Learn how to use TV and BYTSCL, including *all* the BYTSCL
> keywords. You will be undermining a lot of the work we are doing
> here if you use TVSCL.
>
> (And if you *really* want to work in IDL, get TVIMAGE or IMGDISP
> from one of the usual places on the Internet. You don't want to
> be using TV either. :-)
>
>> Then, I've got a second array that I've
>> determined the % of a given mineral per pixel (say from 0 to 1 or 0 to
>> 100 either way you want to look at it). I can tvscl the % mineralogy
>> no problem in a window on its own (colored or grey scale). But what
>> I'd like to do is to overlay the original image that is tvscl, with
>> another tvscl (which is the % mineralogy) without effecting the
>> original image. So, it is similar to highlighting the area of the
>> image that fit my criteria (like above), but now I'd like it to
>> visually show the areas with a higher and lower % of that mineral as
>> well. The first image would be grey scale, the second overlayed image
>> probably in color. You may think that the entire image would be
>> colored, but no. Because I've already picked out pixels that fit
>> another geochemical criteria first. So, I only have ~20% of the
>> original image to cover. If I display the second image alone, the
>> observer has no context for what they are looking at. But, if I over
>> lay it on the first image, that will provide the context. That's what
>> I'm shooting for.
>
> Have a look at this article, I think this describes what you are
> after:
>
>   http://www.dfanning.com/color_tips/color_overlay.html

Hi,

In this case, I'd argue for a different approach. Assuming that the OP
has

two images im1 (with color scale 1) and im2 (with color scale 2) and a ROI, such that the goal is to have a plot of im1 (with colors 1) outside
the ROI and im2 (with colors 2) inside the ROI, the he could proceed in the following way:

1) convert im1 + color scale 1 to a true color image (call it im1true)
2) convert im2 + color scale 2 to a true color image (im2true)
3) substitute pixels of im1true inside roi with corresponding values from im2true
4) device,/decomposed
4) tv,im1true,/true

That should not be too hard to implement... here's an example (not polished
nor optimized, just a hint)

```
;example

;create 2 images
im1=dist(256,256)
im2=rebin(findgen(256),256,256)

;define ROI
ind=array_indices(im1,where(im1 GT 100))

;black-white is col scale for im1
loadct,0
tvlct,r,g,b,/get
im1truecol=im2truecol(im1,r,g,b);convert to true color

;red temp scale for im 2
loadct,3
tvlct,r,g,b,/get
im2truecol=im2truecol(im2,r,g,b);convert to true color

;assign roi pixel from 2 to 1
FOR i=0L,n_elements(ind)/2-1 DO BEGIN
   im1truecol[ind[0,i],ind[1,i],*]=im2truecol[ind[0,i],ind[1,i] ,*]
ENDFOR

device,decomposed=1
tv,im1truecol,true=3
```

using the function (again, this is a bit rough and is not polished)

```
FUNCTION im2truecol,im,r,g,b
s=size(im)
sx=s[1]
```

```
sy=s[2]
n=n_elements(r)
maxim=max(im)
minim=min(im)
im=(im-min(im))/(max(im)-min(im))*n
im3col=[[[reform(r[reform(im,sx*sy)],sx,sy)]], $
        [[reform(g[reform(im,sx*sy)],sx,sy)]], $
        [[reform(b[reform(im,sx*sy)],sx,sy)]]]

return,im3col

end
```

Ciao,
Paolo

>
> Cheers,
>
> David
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming: http://www.dfanning.com/
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")