
Subject: 3D array imaging with different colors

Posted by [tackmeister](#) on Thu, 28 Feb 2013 16:14:35 GMT

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

Hi all,

The past few months I've been breaking my head on a particular problem. I try to create RGB images using the image function. My array to plot is three dimensional, an example image would be:

```
a=fltarr(500,500,3)
for i=0,499 do if i lt 300 and i gt 200 then a(i,*,0)=i
for i=0,499 do if i lt 300 and i gt 200 then a(*,i,0)=i
for i=0,499 do if i lt 100 and i gt 0 then a(i,*,1)=i
for i=0,499 do if i lt 100 and i gt 0 then a(*,i,1)=i
for i=0,499 do if i lt 500 and i gt 400 then a(i,*,2)=i
for i=0,499 do if i lt 500 and i gt 400 then a(*,i,2)=i
```

I plot this array using:
graph = image(a)

The problem now is that every pixel with value (0,0,0) as (R,G,B) will be plotted black. Personally, I'd prefer it if the scaling would occur from white to black instead of the other way around (by default).

I tried to change the color table used, yet this doesn't work. The only thing that does help is by inverting the bytsclcd array, yet this inverts the entire color table and gives strange looking color. Using the tv procedure instead of the image function also doesn't help.

In any case, does anyone of you have an idea how I can image my 3D array as an RGB image, scaled from white (lowest intensity) to black (highest intensity)?

Thanks in advance.

Subject: Re: 3D array imaging with different colors

Posted by [tackmeister](#) on Fri, 01 Mar 2013 15:03:35 GMT

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

Op vrijdag 1 maart 2013 15:49:26 UTC+1 schreef David Fanning het volgende:

- > If you just want to change the background color. Find all the pixels
- >
- > that have zeros in all three color channels and change the value of
- >
- > these pixels to 255 in each color channel. Do the opposite to change
- >
- > white pixels to black.

>
>
>
> I doubt this will be satisfying to you, because normally in a "mixing"
>
> situation like this, there will be a lot of pixels that are "almost"
>
> black (e.g., [0,1,4]). These will not be found and won't be changed.
>
>
>
> I think it will be easier in your situation to just learn to love black
>
> backgrounds. You will sleep better at night. :-)
>
>
>
> Cheers,
>
>
>
> David
>
>
>
>
> --
>
> David Fanning, Ph.D.
>
> Fanning Software Consulting, Inc.
>
> Coyote's Guide to IDL Programming: <http://www.idlcoyote.com/>
>
> Sepore ma de ni thue. ("Perhaps thou speakest truth.")

Aye, I was afraid this would be the final idea. Changing black to white and the other way around does not work perfectly for the reasons you supply yourself. I suppose I could work with a cut-off somewhere that allows for more black-ish values to turn white and more white-ish values to turn black.

But I suppose the 'loving the black backgrounds' would be the best idea indeed. I'll try to work on it ;)

(if in the meantime of course someone does know a better way of doing these things, feel free to share. My latest idea was the following:

```
;define color table variable for red, green, blue signals:
```

```
table = [62,53,49]
```

```
;plot the R, G, B images one after the other using transparency:
```

```
for i=0,2 do graph=image(a[*,*],i,rgb_table=table[i],/overplot,transparency=67)
```

This works as far as I see, though there is no 'color mixing' and the image looks kind of 'smoky' because of the transparencies. So overall the black background images are to be preferred.)
