Subject: Re: 3D array imaging with different colors
Posted by tackmeister on Fri, 01 Mar 2013 11:39:18 GMT
View Forum Message <> Reply to Message

Op donderdag 28 februari 2013 18:15:21 UTC+1 schreef David Fanning het volgende:
> David Fanning writes:
>
>
>
>> You seem to know what colors you want. Just find the RGB value for that
>
>> color and assign those values to the pixels you want to be that color.
>
>> Simple as that.
>
>
>
> You want something like this:
>
>
>
> a = fltarr(500,500,3) + 128
>
> color = [0, 0, 255] ; Blue
>
> a[50:99,200:249,0] = Replicate(color[0],50,50)
>
> a[50:99,200:249,1] = Replicate(color[1],50,50)
>
> a[50:99,200:249,2] = Replicate(color[2],50,50)
>
> color = [255, 255, 0] ; Yellow
>
> a[200:249,50:99,0] = Replicate(color[0],50,50)
>
> a[200:249,50:99,1] = Replicate(color[1],50,50)
>
> a[200:249,50:99,2] = Replicate(color[2],50,50)
>
> color = [255, 0, 0] ; Red
>
> a[400:449,250:299,0] = Replicate(color[0],50,50)
>
> a[400:449,250:299,1] = Replicate(color[1],50,50)
>
> a[400:449,250:299,2] = Replicate(color[2],50,50)
>
> graph = image(a)

>
>
>
> 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.")

Hi David,

Thanks for your answer! However, I'm afraid I depicted my 'problem' too simple and/or I don't fully understand your latest comment.

The actual array I use ranges from much more than 0-400. In fact it is an intensity image, which can range from 0 to a few million counts. Each 2D array (in the 3D array) represents the intensity distribution of a different element in a sample (say Zn, Fe and Cu). Before plotting I would scale the values to give values between 0 and 255 using the bytscl function, so this makes the pixel values in the correct color range.

Now I want to create an image which shows the Zn distribution in red, Fe in green and Cu in blue, in such a way that when all three elements (Fe, Cu and Zn) are present in the same pixel, this pixel is plotted as black. When none of the elements are present in a pixel, it should be plotted as white. When only Fe and Zn are found in a pixel, but no Cu, I'd want the pixel to show the color equal to 155*green+200*red if the scaled intensity of Fe=155 and the scaled intensity of Zn=200 (does this make sense?).
In the end I'd want an image that looks like this:
http://ars.els-cdn.com/content/image/1-s2.0-S000326701000154 6-gr7.jpg (this is actually an image from one of the researchers in the same group as I work in) but instead of scaling from black to white, I want it inverted but I want to keep the nice red, blue and green colors. (because I tried reversing my image array, which runs from 0-255, yet this drew the image in the pink, yellow and light blue, which I do not like very much).

If I understand your first post correctly, by using a 3D array in the image function I define the colors directly by their RGB channel, so if I want to get a different color output I'd have to manipulate RGB channel seperately and change it to the color I want? Is there a way you could

suggest me to do this automatically so I get the result I'd like?

Thanks again for your fast response, and I hope to get your (or someone else's) advise again.
