
Subject: Re: From tvscale to cgImage, puzzling feature in the plot

Posted by [DAVIDE LENA](#) on Tue, 05 Mar 2013 23:37:03 GMT

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

Actually, in the colorbar, I see a white mark that shouldn't be there. Indeed the white pixels correspond with the value marked by the white mark.

Subject: Re: From tvscale to cgImage, puzzling feature in the plot

Posted by [David Fanning](#) on Tue, 05 Mar 2013 23:59:22 GMT

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

Gurdulu' writes:

> I am trying to rewrite a code that was using tvscale to a new one using cgImage and other new functions.

> While using cgImage I run into a problem that was absent into the previous version of my code and I can't figure what is going on. This is what happens:

> I am making "channel maps" for a galaxy (so I read a 'cube' with two spatial dimensions and a frequency dimension, from that I read a spectrum with an emission line, I slice the emission line and make a plot so that each plot shows the distribution of gas with a given velocity. The entire thing goes through a for cycle):

>

> nx = x dimension of the image (spatial)

> ny = y dimension of the image (spatial)

>

> ; Print on the screen:

> erase

> cgIMAGE, rebin(alog10(image),8*(1+nx),8*(1+ny)), POSITION = p,

/KEEP_ASPECT_RATIO,

> MINVALUE = -2, /SCALE, /AXES, AXKEYWORDS={TICKLEN:-0.0075}

> cgCONTOUR, image, COLOR = 'black', NLEVELS=4, LABEL=0, /OnImage

> cgColorbar, RANGE = [-2,0], position = [0.9, 0.1, 0.915, 0.9], /VERTICAL, /RIGHT,
FORMAT='(F4.1)'

> wait, 0.5

>

> The first plot looks fine. All the other plots show some sort of isocontour of white pixels. This 'isocontour' changes with the keyword MAXVALUE in cgImage. Though all the data in the image are within reasonable values. It doesn't seem that the problem relates to the presence of cgContours or cgColorbar, nor rebinning. It shows up with whatever image I plot. Still it seems related to the presence of the FOR cycle. Something happens after the first run.

>

> I hope I have been decently clear.

> Any suggestion?

I would do two things. (1) Immediately update your Coyote Library and make sure the new one is the ONLY one on your IDL path. See suggestions here:

http://www.idlcoyote.com/code_tips/fixcoyoteprogram.php

And, (2) bracket all your image display code with these two commands (one before the graphics code and one after):

```
SetDecomposedState, 1, Current=thisState  
... graphics commands here.  
SetDecomposedState, thisState
```

Normally, this is caused by using `cgColor` incorrectly when using indexed color mode. That is probably what is happening here. But I can't tell if it is because you are using really old Coyote Graphics routines or you have more code somewhere that you aren't showing us. In any case, getting the heck out of indexed color mode will solve your problems. :-)

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.")

Subject: Re: From tvscale to cgImage, puzzling feature in the plot

Posted by [DAVIDE LENA](#) on Wed, 06 Mar 2013 00:09:00 GMT

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

You wizard! The following worked.

```
SetDecomposedState, 1, Current=thisState  
... graphics commands here.  
SetDecomposedState, thisState
```

Anyway the libraries were already updated at the last available version and all the remaining code is just doing operations that do not deal with colors or plotting.
Anyway, thank you very much.

Subject: Re: From tvscale to cgImage, puzzling feature in the plot

Posted by [le.davide](#) on Thu, 07 Mar 2013 15:30:38 GMT

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

David, it turns out that you were right on multiple levels: there was some code that I didn't show and actually created the problem, right after the plot:

```
XYOUTS, p[0]+0.13, p[3]-0.08, vel, charthick=1.,charsize=2, Font = -1, /normal, color =  
cgColor('white')
```

I suppose that the way I chose the color messes up something. There is a solution specific to this problem? I mean, other than doing what you already suggested above - which works - can I just chose the color in another way that doesn't create the problem?

Thanks in advance.

Subject: Re: From tvscale to cgImage, puzzling feature in the plot

Posted by [David Fanning](#) on Thu, 07 Mar 2013 15:44:46 GMT

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

Gurdulu' writes:

> David, it turns out that you were right on multiple levels: there was some code that I didn't show and actually created the problem, right after the plot:

>

> XYOUTS, p[0]+0.13, p[3]-0.08, vel, charthick=1.,charsize=2, Font = -1, /normal, color =
cgColor('white')

>

> I suppose that the way I chose the color messes up something. There is a solution specific to this problem? I mean, other than doing what you already suggested above - which works - can I just chose the color in another way that doesn't create the problem?

Sure.

```
cgText, p[0]+0.13, p[3]-0.08, vel, charthick=1.,charsize=2, $  
Font = -1, /normal, color = 'white'
```

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.")

Subject: Re: From tvscale to cgImage, puzzling feature in the plot

Posted by [David Fanning](#) on Thu, 07 Mar 2013 15:57:47 GMT

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

David Fanning writes:

```
> cgText, p[0]+0.13, p[3]-0.08, vel, charthick=1.,charsize=2, $  
> Font = -1, /normal, color = 'white'
```

By the way, I wouldn't set the FONT keyword on Coyote Graphics routines. The font type is normally determined in Coyote Graphics routines by the particular device you are using. So Hershey fonts on the display and true-type fonts in PostScript. Here you will never get this text to look great in PostScript output (or, by extension, on high-resolution raster output, such as PNG files).

Also the character size and thickness is adjusted properly for the different between the device and PostScript output, so you rarely have to bother with this either. (I'm talking about setting up your PostScript output with PS_Start.) All thicknesses are multiplied by three, for example, to give the proper "weight" to PostScript lines.

Also character size is adjusted for different devices with cgDefCharSize. This is about the only thing I sometimes find myself adjusting in programs, and then only to make the actual character size smaller for legends and the like. I usually write my code like this:

```
textSizeAdjustment = (!D.Name EQ 'PS') ? 0.65 : 0.85  
textSize = cgDefCharSize()*textSizeAdjustment  
cgText, ..., Charsize=textSize
```

This preserves the character size differences between the display and a PostScript file, and makes the PostScript output a more faithful representation of what I see on the display.

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.")