

---

Subject: From tvscale to cglImage, puzzling feature in the plot  
Posted by [DAVIDE LENA](#) on Tue, 05 Mar 2013 23:21:45 GMT

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

---

Hi IDL gurus,

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

While using cglImage 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 cglImage. 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 mage 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?

---

---

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

Posted by [David Fanning](#) on Thu, 07 Mar 2013 16:03:50 GMT

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

---

David Fanning writes:

- > 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.

I suppose I should mention that Coyote Graphics routines also take into account the different default character sizes between Windows and UNIX computers. So `cgDefCharSize()`, which all Coyote Graphics routines use to determine character sizes, distinguished between Windows, UNIX, and PostScript devices, and also whether you are using `!P.Multi` or not.

This can only be accomplished if you do NOT set `!P.CharSize`.

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.")

---