Subject: Postscript dump to a Laserwriter
Posted by scowen on Thu, 17 Dec 1992 21:03:03 GMT
View Forum Message <> Reply to Message

Hi,

OK this seems the simplest thing in the universe right? Wrong.

Let me explain what I've been going through the last few days, and maybe someone out there with a little more experience than I can help me.

Given that I currently have an image displayed into a window with a given color table loaded (other than b/w linear) and that I have modified the stretch to reveal more detail, this is what I'd like to do. I would like to dump the window contents to a Postscript environment (set_plot, 'ps') with the stretch being preserved.

1st pet peeve - tvrd() will only dump the screen displayed image - ie. if you've created a display widget with scrollbars, because the image is too big for your display and you need full resolution so rebin() is not an option then tough! You get an array returned that is the size of your displayed array, but only the visible part will be filled in. I mean what is the point of including such useful things as scrollbars if you can't bloody access the image data off screen - why doesn't tvrd() dump the window buffer instead of doing a screendump?

Anyway, to continue. If I use a 'device, /color' inside the PS setup, then we're off to the races, I can reload the tvrd() buffer and use tvlct with the colors common block to reload the current stretch. However, the resulting PS file will not print on anything but a color PS printer since it puts the colormap in as a /COLORTAB call. OK, so I turn color off using 'device, color=0'. This has the remarkable effect of producing PS that will print on a Laserwriter, BUT (and this is a big one) the colormap used is the default setting of the b/w linear, with all stretch information lost. In addition it doesn't even do a b/w conversion of the existing colortable (eg. Rainbow) but instead reloads the b/w linear.

Now, after all these rantings there is a way to make this work, but it causes something else to stop working. The above methodology allows you to annotate the display, like add contours to an image, or somesuch. Then you ca just dump the window contents using tvrd() and you can print the output. This however breaks the color common block, with the r_curr, etc. being reset so that a tvlct reloads the original colormap and not the strectched one.

The briefly mentioned solution to the Laserwriter PS problem is to actually use tvscl and not use tvrd at all. This way, when you tvscl inside PS the same colormap is used, and you can switch to bits_per_pixel=4 and get a decent b/w rendition of your color stretch. BUT, again, if you have used

contour in a noerase mode, then the colormap is erased and a tvscl renders the default map again.

So, after this lengthy explanation I have several request/questions:

1. Given a color strectched image with contours overlaid, how can I produce *either* color or laserwriter PS from a tvrd() call (ignoring the screen-dump bugaboo)?

2. Alternatively, using the tvscl method, is there anyway to stop the colormap in colors (common) from being erased by the contour call, so we could reload the strectched version of the colortable before tvscl is called. And then (long winded I know) redo the contours inside the PS setup?

3. Does RSI plan to fix this rather annoying feature in tvrd() since it doesn't appear to be a terribly uncommon feature for such a function, to grab the data that is off screen instead of dumping the actual screen buffer itself?

Thanks for your (considerable) time in reading this. I hope someone out there can help.

--

Paul A. Scowen INTERNET: scowen@wfpc3.la.asu.edu
Department of Physics & Astronomy uk1@spacsun.rice.edu
Arizona State University Tel: (602) 965-0938
Tempe, AZ 85287-1504 FAX: (602) 965-7954
