

Paul Levine writes:

>  
> On 2013-08-14 03:51:04 +0000, David Fanning said:  
>  
>> Paul Levine writes:  
>>  
>>> Also, I am quite interested in your take on the PNG files I wound up  
>>> with, where the pixel size of the files was quite different from the  
>>> xysize and yysize specified, and some axis titles and labels were cut  
>>> off. I don't how related this is to the PDF issue, so if you can only  
>>> deal with one of these issues at a time, then perhaps you need to pass  
>>> the other one off to Coyote and see where he gets with it ;-)  
>>  
>> Yeah, I'll talk to Coyote about this, but he has me drinking gin and  
>> tonics tonight to celebrate something or other. It will have to be in  
>> the morning. Oh, I remember, birthdays! We'll sort it out in the  
>> morning. :-)  
>>  
>> Cheers,  
>>  
>> David  
>  
> Ah, well, in that case, happy birthday! I hope for your sake that you  
> don't see this message until tomorrow morning, but once you do, I have  
> something interesting to report on PNG conversion. It seems that to a  
> certain extent, cgps2raster creates PNG files of a certain size that is  
> determined by the aspect ratio rather than the wxsize/wysize or the  
> cgControl resize keyword. So, for example, any aspect ratio of 4:3  
> will produce a PNG of 742 x 556 pixels, and any aspect ratio of 1:1  
> will prodice a PNG of 573 x 573 pixels.  
>  
> So when I appended my code with the following lines:  
>  
> cgControl, Resize=1, Output='Resize\_1x1.png'  
> cgControl, Resize=50, Output='Resize\_50x50.png'  
> cgControl, Resize=800, Output='Resize\_800x800.png'  
> cgControl, Resize=8000, Output='Resize\_8000x8000.png'  
> cgControl, Resize=25000, Output='Resize\_25000x25000.png'  
>  
> each of the resulting PNG files was 573 x 573 pixels.  
>  
> Now, here's where it gets really interesting. I said "to a certain  
> extent" before because once the size is above a certain threshold, then

> a different sized file is produced. Resize=32767 and any number below  
> gives me the 573 x 573, but Resize=32768 and any number above that  
> produces a PNG file that is 742 x 390 pixels . Of course, it can't be  
> a coincidence that the threshold I did pinpoint is the 15th power of 2.  
> Just for laughs, I decided to try finding out whether there was an  
> even higher threshold. So I tried Resize=2.14748e+09 ( $2^{31}$ ), which was  
> still 742 x 390, then I tried to go even higher than that and ended up  
> getting a "Floating illegal operand" error from cgPlot. So I guess  
> cgPlot is not able to handle more than about 2.14748 x 2.14748 billion  
> pixels ;-P  
>  
> Tomorrow, I will hopefully have access to a windows machine, which I  
> can use for testing

Well, OK, I think I can explain this. Even with a slight headache and a thick tongue.

The aspect ratio is key because the way the PNG file gets made is with a PostScript intermediary file. To maintain the highest resolution, I try to create the largest possible "window" on the PostScript page that maintains that aspect ratio. This "high resolution" PostScript file is then "reduced" in size or resolution when the raster file is created by ImageMagick. The default reduction is to 25% of its original size, which gives me a PNG image, say, that is about the size of a normal IDL graphics window when I open it with my image software (Hypersnap, typically). The amount of size reduction is user controlled by means of the IM\_RESIZE keyword in cgWindow\_SetDefs and cgControl.

Note that function graphics raster files are always HUGE. This is because they don't apply this size reduction as their default. You have to specify it as a keyword. (One of many things that bug me about function graphics.) Coyote Graphics routines produce comparably-sized files if you set the IM\_RESIZE keyword to 100.

[http://www.idlcoyote.com/cg\\_tips/compcont.php](http://www.idlcoyote.com/cg_tips/compcont.php)

To give a little more control over the absolute dimensions of the raster file output, I have added a WIDTH keyword to PS\_END (configurable with the IM\_WIDTH keyword for cgWindows). This allows the user to specify a specific width resolution for the raster file. A width of, say, 800 creates a PNG file with a width of 800 pixels and a height based on the aspect ratio of the intermediary PostScript "window". This is handy for me when I am creating PNG output to put on my web page, for example, because I can get the width set exactly as I want it.

I honestly don't know why the resolution of the PNG files change at a certain huge window size. But, I presume it has something to do with the way I compute the largest PostScript "window". I'm probably using short

integers to do the computations, since the possibility of someone wanting a graphics window of 32676 pixels never occurred to me. ;-)

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.")

---