Subject: Re: Incomplete ouput PNG files.
Posted by David Fanning on Wed, 14 Dec 2011 22:21:40 GMT
View Forum Message <> Reply to Message

Mark Piper writes:

> This is a slightly different workflow, but could you please try setting
> the BUFFER keyword in your call to IMAGE? E.g.,
>
> p = image(data, /buffer)
> p.save, 'this_image.png'
> p.close
>
> The graphic will be rendered in an offscreen buffer. I have a hunch that
> this may help, since this feels like a tricky (to me, at least) X server
> issue.

I was curious to see how Coyote Graphics output would
stack up against the output from these function graphics
routines. But I wanted to be able to compare apples
to apples, so I spent some time today modifying the
Coyote Graphic routines so that I could control
the output file parameters, and in particular, the
resolution of the output.

This is now done with cgWindow_SetDefs, just like
it is for cgWindow. In my first comparisons, I noticed
that the function graphics output was a bit darker
than the Coyote Graphics output, so I defined a new
keyword for PS_START, called DEFAULT_THICKNESS so that
I can set the default line and character thickness for
the PostScript output. I set the default to 3 to better
match the function graphics output.

Anyway, you will need an updated Coyote Library to run
the program described, if you want to play around with this:

http://www.idlcoyote.com/programs/zip_files/coyoteprograms.z ip

This is tagged release 1.5.1, if you are using the Subversion
repository.

So, here is the program. I'm doing a simple plot command and
saving the data as JPEG, PNG, and encapsulated PostScript files.
(Coyote Graphics routines actually produce landscape PostScript
files, which function graphics commands do not, so I am using
encapsulated PostScript for my comparisons. Both will produce
encapsulated output in Portrait mode.) I've saved the files

at 600 dpi, 300 dpi and 75 dpi.

I was careful to make sure I was using the same size window
in both cases, 640 in X and 512 in Y.

In general, I can't really tell much difference in the output.
The title is set too close to the plot, but that has always
been the case in direct graphics. That is about the only
difference that really jumps out at me.

A couple of odd things. The PostScript files are all the
same size at every resolution. They are 11KB for Coyote
Graphics output and 9 KB for function graphics output.
Here is a table of values in KM. The size values are
a comparison of the output. You can see that Coyote
Graphics routines are consistently larger in dimensions,
but smaller in total size. I don't know how to account for
this. In any case, the visual output is comparable so
I assume this is just a different way of setting the
resolution. The XSIZE and YSIZE dimensions are for the
JPEG file in every case, but the comparable PNG file
has the same dimensions.

	EPS	JPEG	PNG	XSIZE	YSIZE
cg75	11	39	63	717	573
fg75	9	39	33	667	534
cg300	11	227	46	2867	2292
fg300	9	254	165	2669	2135
cg600	11	568	131	5733	4583
fg600	9	736	379	5339	4271

I guess the bottom line is that I am EXTREMELY happy
with the performance of Coyote Graphics in this
comparison. Not only are my routines faster, but the output
I care about is essentially identical to the output
from function graphics routines. As an added bonus,
my output files are significantly smaller at high
resolution. I don't know why this would be the case.

Here is the code I used, if you want to try this for
yourself:

http://www.idlcoyote.com/misc/compare_resolution.pro

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")
