Subject: Re: Pixels per Inch
Posted by JD Smith on Tue, 16 Dec 2003 18:49:29 GMT
View Forum Message <> Reply to Message

On Tue, 16 Dec 2003 10:03:36 -0700, Pepe wrote:

> Dear JD et al,
>
> Thanks for your help.
>
> I was wondering if it would be possible to read all the 'PNG' images I
> already created (using the "read_png" routine) and then converting them
> to 'PostScript'. I could thus set the desired resolution while creating
> the new 'PostScript' image. Would this approach work ?
>
> My real problem is that I want the new images to have the same "physical
> size" as before, but with increased resolution. How can I do this ? Do I
> set the resolution using the 'xsize' and 'ysize' keywords in "device" ?
>
> I had a look at the IDL help pages and came up with an example that made
> me write the following simple program. The problem is that I don't know
> how to keep the "physical size" while increasing the resolution.
>

When you display image data in postscript, it just sticks a bitmap version of the image inside. The normal "vector scaling" properties of postscript, which allow it to look good at any physical (i.e. print) resolution don't apply to bitmap data, since postscript has no way to increase the amount of detail in the image (it just makes the image pixels bigger). This is in contrast to things like lines and letters, where the postscript contains a *model* for their appearance. Instead of a series of dots at specific places, postscript, and other vector formats, contain a model roughly equivalent to "line with width 4 pixels and position y=22x+433". Then, depending on the final resolution at which the Postscript is being printed, the line equation is used to "fill in the dots".

The only way to improve *real* resolution of an image, i.e. the smallest detail which can clearly be seen (think of the little numbers at the bottom of an eye chart), is to increase the image size in pixels. It's best not to think in terms of a "physical size" for an image. The "physical size" of an image on your display depends on how many pixels per inch you are displaying, since images are usually shown pixel by pixel.

That said, if, for some reason, all you want to do is give the image readers the suggestion of what physical size it might be appropriate to render the image at (remembering that many image readers ignore

this), you could use postscript to do this, or you could encode it in
the 'pHYs' block of the PNG using pngcrush. Probably the postscript
route would be more likely to succeed. Using device,XSIZE=,YSIZE=
should do what you expect (possibly with the addition of /INCHES).
You'll also need the XSIZE,YSIZE,INCHES keywords to TV to get the
image to display at the right size (in inches). Just be aware that
all this fancy footwork is doing is specifying how to re-scale your
image. You could achieve the exact same thing by re-scaling it
yourself before printing (and you could use a fancier interpolation,
if you liked).

Here's an example of how postscript can display different sized images
as the same "physical size".

```
IDL> a=dist(100)
IDL> set_plot,'PS' & device,FILENAME='small.eps',/encap,xsize=5,ysize=5,/INCHES
IDL> tvscl,a,0,0,xsize=5,ysize=5,/INCHES
IDL> device,/CLOSE & set_plot,'X'
IDL> a=dist(1000)
IDL> set_plot,'PS' & device,FILENAME='large.eps',/encap,xsize=5,ysize=5,/INCHES
IDL> tvscl,a,0,0,xsize=5,ysize=5,/INCHES
IDL> device,/CLOSE & set_plot,'X'
```

When you display or print these, you'll get the same physical size
(roughly 5 inches, pretty close on your printer, more variable on your
display). The second will look much sharper, since it started with a
much large image, and is therefore encoded at 200dpi. You can even go
extreme:

```
IDL> a=dist(10000)
IDL> set_plot,'PS' & device,FILENAME='huge.eps',/encap,xsize=5,ysize=5,/INCHES
IDL> tvscl,a,0,0,xsize=5,ysize=5,/INCHES
IDL> device,/CLOSE & set_plot,'X'
```

and you'll see a large increase in the file size:

```
turtle% ls -l *.eps
-rw-rw-r--    1 jdsmith  jdsmith  100835345 Dec 16 11:38 huge.eps
-rw-rw-r--    1 jdsmith  jdsmith    1010340 Dec 16 11:35 large.eps
-rw-rw-r--    1 jdsmith  jdsmith      12085 Dec 16 11:35 small.eps
```

But when you display huge.eps, it won't look any better than
large.eps. Why? Because 200dpi is already more than your display can
show; 2000dpi is *way* more than your display can show, and much more
than most printers can accurately reproduce. It will, however, take
much longer to load, as it goes through all those pixels throwing away
19 out of 20 of them.

JD