Subject: Re: Pixels per Inch
Posted by R.Bauer on Mon, 15 Dec 2003 13:52:18 GMT
View Forum Message <> Reply to Message

Pepe wrote:
> Good morning,
>
> I have created a PNG image using the "write_png" routine. I have
> subsequently been asked to increase the resolution of this image from
> 96 pixels per inch to 300 pixels per inch, how can I do this ?
>
> I would be very grateful if someone could point our to me how to
> determine the resolution of a PNG image, and how to vary it (if
> possible) ?
>
> Thanks in advance for the help.
>
> Best Regards,
> Pepe

Dear Pepe

you should write into a postscript file and then you can use the pstoimg
routine from the latex2html package or convert from ImageMagick to
convert from ps to png.

With both you can set the resolution.

Reimar

--
Reimar Bauer

Institut fuer Stratosphaerische Chemie (ICG-I)
Forschungszentrum Juelich
email: R.Bauer@fz-juelich.de
 ---------- -------
 a IDL library at ForschungsZentrum Juelich
 http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_lib_intro. html
 ========== =======

Subject: Re: Pixels per Inch
Posted by JD Smith on Mon, 15 Dec 2003 18:00:46 GMT
View Forum Message <> Reply to Message

On Mon, 15 Dec 2003 04:45:50 -0700, Pepe wrote:

> Good morning,
>
> I have created a PNG image using the "write_png" routine. I have
> subsequently been asked to increase the resolution of this image from 96
> pixels per inch to 300 pixels per inch, how can I do this ?
>
> I would be very grateful if someone could point our to me how to
> determine the resolution of a PNG image, and how to vary it (if
> possible) ?
>
> Thanks in advance for the help.

It depends on how many "inches" you intend to output the image to. This is a basic and common mis-understanding of how image resolution works. A bitmap image like a PNG file has a specific size in pixels, nxm say, but does not have an associated "physical size". The physical size of this image depends on the device used to show it: on my monitor, for instance, it would be around n/100 x m/100 inches, since I display around 100dpi. On a 1200dpi printer, it would be 12 times smaller. So what about that nice 1200x800 digital picture you printed on a 1200dpi printer: it certainly wasn't 1 inch x 2/3 inch! This is the main source of confusion: the printer or printer driver resized your image to 1200dpi by interpolating or rebinning it. Just because it was printed on 4x6" photo stock doesn't mean it actually contains 4800 x 7200 pixels of information: most of it was manufactured by the printer (another semi-relevant complication is that color printers use patterns of solid "dots" of color to build up images: much different from a monitor pixel which can display any color by varying the RGB intensity, so 100dpi on screen != 100ppi on a printer).

Further complicating the issue is the insistence of many people to refer to, e.g., 300dpi JPG images: this usually means they've assumed some "natural" size of the image in inches, and scaled the pixel size to that. And indeed many bitmap formats, including JPG and PNG, add to the confusion by supporting an image header which specifies the DPI resolution: but this does not do *anything* to the actual pixels or the amount of detail in the image, it just makes a "suggestion" to programs using the image regarding how large to display or print it (i.e. how much interpolation/rebinning should be done). Many programs ignore this information altogether.

On the other hand, vector data like postscript files *do* have a physical size, but they don't have a pixel size. This is mostly relevant for printers, and the printer resolution enters in determining how fine accurately-printed postscript features are (think of two thin lines quite close to each other). To display or print postscript, the program or device needs to have at least an

approximate understanding of the resolution of the display or printer.

In short, I suspect what they really meant is triple the pixel size of the image, and they will print it at the same physical size. If they really just want you to change the header value "96" to "300", I think PhotoShop will do this for you, but remember that this does nothing to the actual information and resolution present in the image.

JD

Subject: Re: Pixels per Inch
Posted by Craig Markwardt on Mon, 15 Dec 2003 20:20:08 GMT
View Forum Message <> Reply to Message

JD Smith <jdsmith@as.arizona.edu> writes:
[...]
> It depends on how many "inches" you intend to output the image to.
> This is a basic and common mis-understanding of how image resolution
> works. A bitmap image like a PNG file has a specific size in pixels,
> nxm say, but does not have an associated "physical size". The

JD, I'm not sure that's always true. I believe that TIFF images have an inherent (desired) dot pitch. Perhaps other formats do too, but PNGs definitely do not, so your comment is appropriate for the original poster.

Craig

Subject: Re: Pixels per Inch
Posted by JD Smith on Mon, 15 Dec 2003 21:46:17 GMT
View Forum Message <> Reply to Message

On Mon, 15 Dec 2003 13:20:08 -0700, Craig Markwardt wrote:

> JD Smith <jdsmith@as.arizona.edu> writes: [...]
>> It depends on how many "inches" you intend to output the image to. This
>> is a basic and common mis-understanding of how image resolution works.
>> A bitmap image like a PNG file has a specific size in pixels, nxm say,
>> but does not have an associated "physical size". The
>
>
> JD, I'm not sure that's always true. I believe that TIFF images have an
> inherent (desired) dot pitch. Perhaps other formats do too, but PNGs

> definitely do not, so your comment is appropriate for the original
> poster.

Actually, PNG also has an optional header chunk called "pHYs" which gives the pixels per meter. Other image formats have something similar. My main point was that, no matter what you put in the header regarding the "physical size" of the image, this is only a suggestion, and has no bearing on the resolution or detail present in your image. Many programs explicitly ignore these data (especially for on-screen display). You have to separate the notion of resolution as "convenient suggestions for printing and print layout programs" vs. resolution as real physical detail present in the image.

The program pngcrush can add, change or remove the pHYs resolution information from PNG files (http://pmt.sourceforge.net/pngcrush). But, for example, if I take a PNG file and give it resolution 300dpi, 50dpi, and no resolution, the three version display exactly the same in all browsers, and the Gimp.

JD

Subject: Re: Pixels per Inch
Posted by sdj on Tue, 16 Dec 2003 17:03:36 GMT
View Forum Message <> Reply to Message

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.

```
;**********
PRO PNG2PS, in_file
```

```
;Read the PNG file.
img = read_png(in_file, r, g, b)

;Reconstruct the color table.
tvlct, r,g,b

;Display the image in an IDL window.
tv, img

;Find the size of the picture.
s = size(img)

;Take the `png' extension off of the old filename and replace it with
`ps'.
fl = strlen(in_file)
file = strmid(in_file, 0, fl-4)
out_file = file + '.ps'

;Set the plotting device to PostScript.
set_plot, 'ps'

;Use the DEVICE procedure to make the output PostScript
device, bits_per_pixel = 8, /color, filename = out_file, xsize = ??,
ysize = ??

;Write the image to the file.
tv, img

;Close the file.
device, /close

END
;*************
```

Thanks again for the help and excuse me for insisting on this "simple"
problem.

Regards,
Pepe

> On Mon, 15 Dec 2003 13:20:08 -0700, Craig Markwardt wrote:
>
>
>> JD Smith <jdsmith@as.arizona.edu> writes: [...]
>>> It depends on how many "inches" you intend to output the image to. This
>>> is a basic and common mis-understanding of how image resolution works.
>>> A bitmap image like a PNG file has a specific size in pixels, nxm say,

>>> but does not have an associated "physical size". The
>>
>>
>> JD, I'm not sure that's always true. I believe that TIFF images have an
>> inherent (desired) dot pitch. Perhaps other formats do too, but PNGs
>> definitely do not, so your comment is appropriate for the original
>> poster.
>
> Actually, PNG also has an optional header chunk called "pHYs" which gives
> the pixels per meter. Other image formats have something similar. My
> main point was that, no matter what you put in the header regarding the
> "physical size" of the image, this is only a suggestion, and has no
> bearing on the resolution or detail present in your image. Many programs
> explicitly ignore these data (especially for on-screen display). You have
> to separate the notion of resolution as "convenient suggestions for
> printing and print layout programs" vs. resolution as real physical detail
> present in the image.
>
> The program pngcrush can add, change or remove the pHYs resolution
> information from PNG files (http://pmt.sourceforge.net/pngcrush). But,
> for example, if I take a PNG file and give it resolution 300dpi, 50dpi,
> and no resolution, the three version display exactly the same in all
> browsers, and the Gimp.
>
> JD

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

Subject: Re: Pixels per Inch
Posted by William Thompson on Fri, 19 Dec 2003 18:46:35 GMT
View Forum Message <> Reply to Message

JD Smith wrote:
> On Mon, 15 Dec 2003 04:45:50 -0700, Pepe wrote:
>
>
>> Good morning,
>>
>> I have created a PNG image using the "write_png" routine. I have
>> subsequently been asked to increase the resolution of this image from 96
>> pixels per inch to 300 pixels per inch, how can I do this ?
>>
>> I would be very grateful if someone could point our to me how to
>> determine the resolution of a PNG image, and how to vary it (if
>> possible) ?
>>

>> Thanks in advance for the help.

I've come to the conclusion that when people ask for a 300 dpi version of an image, they don't really mean a specific size or header information. All they're asking for is an image with a lot more pixels in it, so that when the image is printed at 300 dpi, it won't be the size of a postage stamp. In other words, "300 dpi" is shorthand for a high resolution (i.e. big!) image.

Bill Thompson

>
>
> It depends on how many "inches" you intend to output the image to.
> This is a basic and common mis-understanding of how image resolution
> works. A bitmap image like a PNG file has a specific size in pixels,
> nxm say, but does not have an associated "physical size". The
> physical size of this image depends on the device used to show it: on
> my monitor, for instance, it would be around n/100 x m/100 inches,
> since I display around 100dpi. On a 1200dpi printer, it would be 12
> times smaller. So what about that nice 1200x800 digital picture you
> printed on a 1200dpi printer: it certainly wasn't 1 inch x 2/3 inch!
> This is the main source of confusion: the printer or printer driver
> resized your image to 1200dpi by interpolating or rebinning it. Just
> because it was printed on 4x6" photo stock doesn't mean it actually
> contains 4800 x 7200 pixels of information: most of it was
> manufactured by the printer (another semi-relevant complication is
> that color printers use patterns of solid "dots" of color to build up
> images: much different from a monitor pixel which can display any
> color by varying the RGB intensity, so 100dpi on screen != 100ppi on a
> printer).
>
> Further complicating the issue is the insistence of many people to
> refer to, e.g., 300dpi JPG images: this usually means they've assumed
> some "natural" size of the image in inches, and scaled the pixel size
> to that. And indeed many bitmap formats, including JPG and PNG, add
> to the confusion by supporting an image header which specifies the DPI
> resolution: but this does not do *anything* to the actual pixels or
> the amount of detail in the image, it just makes a "suggestion" to
> programs using the image regarding how large to display or print it
> (i.e. how much interpolation/rebinning should be done). Many programs
> ignore this information altogether.
>
> On the other hand, vector data like postscript files *do* have a
> physical size, but they don't have a pixel size. This is mostly
> relevant for printers, and the printer resolution enters in
> determining how fine accurately-printed postscript features are (think
> of two thin lines quite close to each other). To display or print

> postscript, the program or device needs to have at least an
> approximate understanding of the resolution of the display or printer.
>
> In short, I suspect what they really meant is triple the pixel size of
> the image, and they will print it at the same physical size. If they
> really just want you to change the header value "96" to "300", I think
> PhotoShop will do this for you, but remember that this does nothing to
> the actual information and resolution present in the image.
>
> JD
