

Photography with Linux

Digital cameras and scanners; image enhancement and storage

Copyright © 2003 Tim Waugh

This article may be used for Red Hat Magazine

Digital cameras are great. When I bought my first digital camera it was a nice toy, and I was impressed with the picture quality—considering it was not a “real” camera—although it was a bit heavy to carry around. At the time, digital cameras were quite a new thing.

They still are comparatively new of course, but now they are real competition for 35mm cameras. My current digital camera has more storage capacity than my desktop computer has RAM. I buy 35mm or APS films less often, and instead find myself taking a CD-ROM to the shop to be “developed”.

When looking for a suitable digital camera to buy, there are some things you should consider. The maximum resolution is an obvious one, and this is measured in mega-pixels. What this really means in terms of actual picture quality is more complicated than it seems on the surface, but suffice to say that at about 4 mega-pixels it is hard (for me, at least) to tell the difference between a 7”x5” 35mm print and one, professionally printed, taken with a digital camera. (It is said our eyes are equivalent to 120 mega-pixels.)

While it is possible to get a digital camera with a detachable lens, so that telephoto or wide-angle lenses can be swapped in, most come with a fixed lens and about 3x zoom magnification. Some will advertise higher magnification “digital zoom”, sometimes twice as high as the optical zoom, and this just means interpolating between the pixels to guess what would have been there. When comparing cameras look at the optical zoom magnification. The very same issue, of “digital resolution” and optical resolution, applies to scanners as well. More about scanning later on, but first let's continue to talk about digital cameras.

Digital Cameras

Different manufacturers use different forms of storage for digital cameras. The only type I have experience with is CompactFlash, but it is worth comparing the other options. For instance, you might want to get a separate computer peripheral for reading the storage media, and of course that will need to be able to read the media format you choose.

Naturally, your most important requirement when choosing a digital camera is being able to fetch the images from it using Linux! As an aside, it is entirely possible to get digital prints from the camera without the aid of a computer by instead using a multi-function “office device”—one of those units that can scan, print, and fax—if it has a card reader or something equivalent or by using a photo printer that accepts a connection directly to your digital camera. The HP OfficeJet devices, for example, print out an index sheet for you to mark with a pen the pictures you want, and then when you scan the marked index sheet it

will print the selected photographs!

Another method of avoiding a computer is to take the camera into a print shop and get them to use one of their storage media readers. Additionally you'll probably want to keep digital copies of the photographs yourself, or change them before getting them printed, and of course you will want to do that using Linux.

A large number of modern digital cameras have USB sockets and, when plugged into the USB port, use the USB "mass storage" language for transferring data. That is, the pictures on the camera appear in a file system just as if they were on a hard disk. All you need to do is right-click with the mouse on the desktop background and select "mass-storage" from the Disks menu. An icon will appear on the desktop, and when you double-click it you will see the files in the file system. Often the pictures will be in a sub-directory within a sub-directory: for my camera they are in /dcim/100pentx/ for example.

Note that the above does not work for all USB mass storage cameras. It depends how the camera describes itself to the computer. If the method described above does not work for you, you can read about how to correct the problem at <http://cyberelk.net/tim/usb-storage.html>.

The Nautilus file manager (the window you get when double-clicking on your home folder icon) can show thumbnails for pictures. If your digital camera has very high resolution and is set to 100% JPEG quality, you might find that thumbnails are not created. This is because by default thumbnails are only created for local image files smaller than 3 megabytes. You can change the size limit for creating thumbnails by choosing Preferences from the Edit menu, switching to the Preview tab and adjusting the option to 5 or 10 megabytes. You can double-click on the thumbnails to enlarge the picture to fill the window, or you can view the directory as an image collection in which case you will be able to see a slide-show of all the pictures.

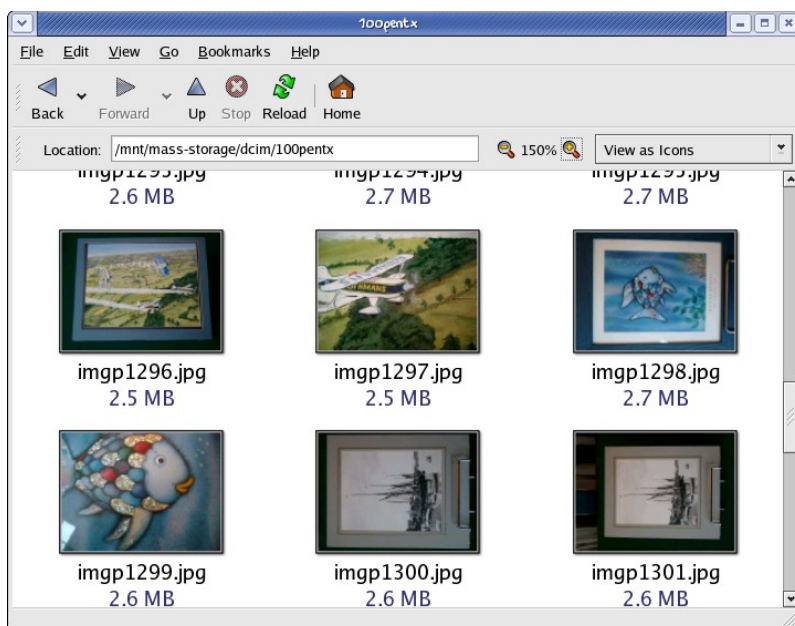


Figure 1 Icon view in Nautilus

USB mass storage is an ideal way to get at the pictures you have taken, but unfortunately not all cameras have USB connections, and even some that do choose to use their own special protocol instead. For these mass storage-unaware USB cameras, and the cameras with serial or other types of connections instead of USB connections, you can use the gPhoto2 camera access program. In fact it is a library, but there are a couple of front-ends available.

One of them is a graphical application called **gtkam** (so-called because its interface uses the GTK+ library). Launch this application from the Digital Camera Tool menu option under Graphics from the main menu. When you first launch it you will need to tell it about your camera by selecting "Add Camera..." from the Camera menu; choose your camera model from the list. For some types of connection you will be able to press the Detect button to find it; for others you will need to tell it which serial port (for example) it is connected to.

After telling it what sort of camera you have, you will see a directory tree in the left pane, and the right pane will have thumbnails of the pictures once you are in the right directory. Unfortunately the **gtkam** interface is not as good as the Nautilus file manager equivalent.

Of course, you can transfer the image files onto the local hard disk and use Nautilus to view them from there. If that is your plan you might find it easier to use the command line **gPhoto2** program to transfer the images from your camera. Basic operation is available with the `--list-files` and `--get-all-files` options, and the manual page (`man gphoto2`) gives more detailed instructions on how to use it.

There is yet another method for transferring pictures to the computer, regardless of whether the camera uses USB mass storage or a different protocol, and that is to use a card reader. These are peripherals you can plug the camera's storage card into, which serve the purpose of reading from them (and in some cases, writing to them). As mentioned before, the HP OfficeJet all-in-one devices have a CompactFlash slot for this purpose, but you can buy devices whose sole purpose is to read camera media USB card readers usually appear as mass storage devices, as discussed above.

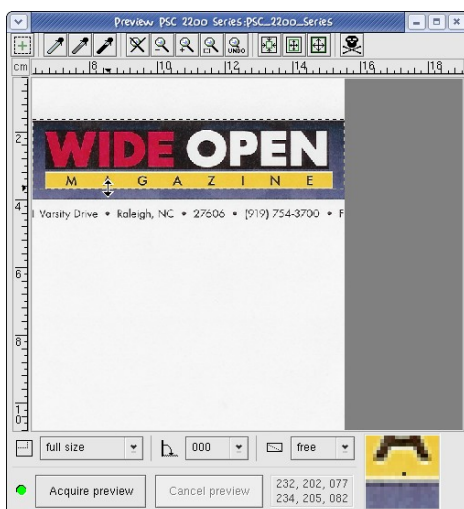


Figure 2 XSane preview window

Scanners

So much for digital cameras; what about scanners? The scanning package supplied in Red Hat distributions is called SANE (it stands for Scanner Access Now Easy), and like gPhoto2 it is a frame-work used by several front-ends. The command line interface for it is **scanimage**, and **XSane** is the graphical version (select "Scanning" from the Graphics menu). If, like me, you often try to touch up or otherwise enhance pictures after scanning them you will probably find The GIMP plug-in part of XSane easiest to use (select "Acquire" from the File menu).

For most types of scanner you will not need to set anything up, and scanning will just work. For scanners connected to the parallel port, or all-in-one devices such as those mentioned earlier, there are things you will need to do first. For OfficeJet devices, you will need to follow the instructions at <http://cyberelk.net/tim/hpoj.html> (which applies to Red Hat Linux 9, and Fedora Core 1; earlier Red Hat Linux versions do not have support at all).

Parallel port scanners and other types that do not work "out of the box" might require adjustments to one of the files in `/etc/sane.d/`. In that directory there is a configuration file for each scanner driver, and each one has a manual page describing the various options it can contain. The first file to check is `dll.conf`, which specifies which drivers are enabled; use `man sane-dll` to view its manual page. If the driver you need is enabled but does not find your scanner, check the configuration file for that driver. For example, for Fujitsu scanners check the `fujitsu.conf` file, and look at the `sane-fujitsu` manual page.

One good tip for getting better picture quality from a scanner is (at the risk of stating the obvious): dust everything. Wipe the dust from the picture you are trying to scan as well as the scanner surface, making sure to use a lint-free cloth. In fact it can even be worth wiping the dust from the scanner lid panel, and I will explain why. On the underside of the glass scanner surface there can be marks or imperfections that, although you may not be able to see them by looking at the scanner, may well show up when scanning something with a lot of light colouration. By making sure the scanner lid panel is clean, you can obtain a scanned representation of the marks on the underside of the glass surface by scanning with nothing under the lid. Using that, you can see where fingerprints or smudges remain on top of the glass, in order to clean them more thoroughly. You could in theory clean up subsequent scans (of actual photographs) by masking off the marks you can't get rid of!

Some scanners even scan in infra-red, which only picks up dust, and this sort of masking off process can be performed automatically by the SANE scanner drivers. For an example of how this is done with Nikon scanners, refer to <http://andreas.rick.free.fr/sane/dustremove.html>.

If there are marks on the picture you are scanning, or other marks on the scanner that you cannot get rid of, you can remove them afterwards in GIMP using the clone tool. This works by copying adjacent bits of the image over the marks, and produces remarkably good results. The only drawback is that it can take some time to clone over all the marks when working with a large image. This tool is so useful that it bears explaining a bit more with an example.



Figure 3 Dust on Tower Bridge



Figure 4 Removing dust using the clone tool

Figure 3 shows part of a scanned image that needs to have some dust removed. I opened this image in GIMP and selected the clone tool from the tool palette (the icon looks a bit like a rubber stamp, or you can press C on the keyboard to select it if you are not sure which icon it is). Then I chose an appropriate brush—the clone tool uses the current brush to copy from the image—in this case a large fuzzy circle. The clone tool copies from a source point to the point you click on, and if you use aligned mode then the source point will follow the destination point as you drag the mouse. In other words, aligned mode copies from a particular offset from the destination point, whereas non-aligned mode copies from a particular image point. I set the clone tool to use aligned mode by double-clicking on the clone tool icon to get the tool options window; in this window you can set other options like opacity and pressure sensitivity, but I rarely need to do that for dust removal. After that I had to select a place to copy from.

The idea is to find an area quite close alongside the dust, or else any area that is the same colour or pattern. With this section of London's Tower Bridge, you can see the repeating diamond pattern along the side: I just copied a section from the left of the dust by control-clicking on the top point of a dust-free diamond shape, and then clicked on exactly the equivalent point on the dusty diamond shape and dragged along the length of the dust mark. You can see this process in Figure 4. The result: no more dust mark.

You can avoid dust altogether by using a digital camera, of course, so long as the lens is clean. But with cameras there is a different problem to avoid, the consequence of catching someone off guard with the flash on: red-eye. Many cameras have a red-eye reduction setting, but it is easy to forget to use it. Removing the red-eye effect is another thing that can be done using GIMP. My technique, before I discovered a better way, had been to select the red eyes using the circle selection tool, then use the Hue-Saturation dialog to mostly desaturate (remove the colour) and slightly darken, so that the eyes ended up dark grey rather than red. I had to be careful to only select the red bits of the eye to avoid making the eye-lids dark grey as well.

A better way is to remove the red from pixels where the red component



Figure 5 Red hat, red eyes



Figure 6 Eyes no longer red

dominates, and a GIMP plug-in has been written to do just this. It is available at <http://fmg-www.cs.ucla.edu/fmg-members/geoff/digicam/> (alongside several other tools for manipulating images). You still need to select the red eyes, but now you do not need to be so careful to exclude eye-lids since they will be unaffected. This plug-in provides two extra “Misc” filters, one for automatic red-eye reduction and one for fine-tuned red-eye reduction. Usually the automatic option just works, but occasionally you will want to adjust the sensitivity to red in which case you need to use the other filter. I used sensitivity -22 to transform the eyes in Figure 5 to those in Figure 6.

Actually before I even started to remove red-eye in the above two Figures, or dust in the Figures before them, I had already used GIMP to enhance them. When a scanned image has large areas of the same colour (sky for example), or a digital camera picture is taken in low light, the image can look speckled: some pixels stand out as being more different from their neighbours than they should. One option would be to blur the image, to ensure that adjacent pixels are similar in colour, but obviously that would make the picture much less sharp in detailed areas. The solution is the despeckle enhancement filter. This filter maintains sharpness in detailed areas to a large degree, and eliminates “noisy” pixels from other areas. Sometimes it gives unwanted results for very small bright areas (such as glints in eyes), and the filter parameters can be adjusted to minimise this at the cost of being less effective at removing bright speckles. My own technique is to select the obvious glint areas, invert the selection, and despeckle the remainder of the image.

Speckles may not just appear on digital camera pictures taken in low light. A digital camera uses a large array of millions of light sensitive devices. If one of them is faulty on manufacture, or breaks down through use, a single pixel in each resulting image will be wrong. It will always report the same colour. It may be black, in which case it will show up on light pictures; it might be bright green, as happened to me. I took the camera back to the shop I bought it from and they repaired it, but there is something you can do even if you discover a similar fault after the warranty has expired. Of course, you could use the clone tool to repair damage, but it would be tiresome to have to do that for every single picture you take.

With defective light sensors in the camera there is an opportunity for automating the process. Since you know where in the image they are going to be, you could program The GIMP to clone over them automatically. In fact, a stand-alone tool has been written for this purpose. It is called **jpegpixi** (JPEG Pixel Interpolator), and it corrects defective pixels by averaging out the colours between its neighbouring pixels. This useful program is available at <http://www.zero-based.org/software/jpegpixi/>. The JPEG image format is lossy,

and programs that manipulate images usually load them, operate on them, and then re-encode them (possibly at lower quality), whereas the author of **jpegpxi** has taken great care to ensure that this technique preserves the quality of picture areas that do not contain defects.

Printing and Sharing Photos

At some stage, once you have the images on your computer you will probably want to print them out. Modern colour printers can give very good photo-quality print output now. The quality of the output depends very much on how the paper you use absorbs the printer ink. You will need to experiment with different paper types to decide which gives the best quality for your printer. If you are looking for true colour representations, bear in mind that different monitors show different colours depending on how much they have been used! I have heard of systems for calibrating the print output by printing test sheets with known colours, scanning them in again to compare them, adjusting the colour correction parameters in the printer driver, and repeating the process until there is no change—but I do not know of any free software for performing this function. Luckily, most of us will not need this level of colour correctness.

The colour correction parameters you are able to set vary between printer drivers. With the printing configuration tool you can see the available options in the Driver Options tab of the dialog window for editing a print queue. The gimp-print driver, for instance, allows you to set the individual magenta, cyan and yellow levels, as well as brightness, contrast and “gamma”, which is to do with how much a given increase in ink darkens the paper. Not all printer drivers give this much control over the colour correction.

The colour can be corrected before it is sent to the printer driver, of course, using The GIMP, the GNU Image Manipulation Program. The “Print” dialog in The GIMP is actually the gimp-print printer driver in disguise. The Image/Output Settings tab has an Adjust Output button for tweaking the colour output. You can tell The GIMP to print the current image as PostScript Level 2. This is useful even if your printer does not understand PostScript, since Ghostscript interprets the PostScript and lets the printer driver convert it to a language the printer does understand—if this is the case for your printer, make sure to remove “-oraw” from the command The GIMP uses to print, using the Setup Printer button.

If you will be printing lots of photographs, pretty quickly you will find out how long the ink cartridges for your printer last! Ink cartridges are expensive, and so is the glossy paper for printing on, and you might find it more economical to use a professional digital photograph printing service. Most places that develop camera films also process and print digital photographs and charge a small amount to do so. I have found that digital prints made this way are both excellent quality, indistinguishable (to my eye) from film-developed prints, and good value compared to printing them myself.

The great thing about having the pictures on your computer is that it is very easy to share them with people. You can email them to friends, or you can upload them to on-line photograph galleries such as Webshots. If you already have your own webpage, and your Web hosting service provides PHP4, you can make your own Web gallery using one of the freely available PHP gallery packages.

The one I use is called, simply, Gallery (<http://gallery.sourceforge.net/>). The galleries you create can contain albums (and sub-albums), and people viewing the photographs in the albums can add comments about them. It is largely customizable and easy to use. Best of all, it has an off-line mode so that you can put an entire gallery onto a CD for safe-keeping, or for sending to a friend. The off-line mode creates a large number of static HTML pages, which can be viewed using any Web browser. In fact I run Gallery on my local machine, just because its off-line mode is good for creating CDs.

After I have taken some pictures with the digital camera, I upload them to the computer to see which ones are any good and which ones are not worth keeping. Then I add the good ones to a Gallery, running locally on the computer. This was easy to set up: install `httpd` and `php`, and unpack Gallery into the right place. Adding captions to the pictures is easy, using the Web interface, and so is rotating the pictures to be the right way up. Once I have a CD's worth of pictures in the gallery I create a CD image using the off-line mode, using a shell script. Here is what the script does:

1. Create an archive of the Gallery directory, excluding the JPEG files. This is so that I can recreate the Gallery from CD if I want to change captions or any other settings.
2. Use **wget** to recursively fetch HTML pages from Gallery in off-line mode. This looks like: `wget -r -k --html-extension -nH --cut-dirs=10 -P html "${URL}?set_offline=true"`— refer to the Gallery documentation for more information about how and why this works.
3. Create a small HTML file that just redirects to the real Gallery `index.html` file (in fact it is called `index.html%3Fset_offline=true.html` because of the way **wget** works).
4. Make an ISO image of the whole lot.

Then I can burn that ISO image to CD and add it to the pile (and another, just in case!). An easier way to create CDs of photographs would be to just put the image files onto CD using the Nautilus file manager. To do this, choose CD Creator from the Go menu, drag the images into that window and then click on the Write to CD button.

However you choose to enhance, share and archive photographs, and whether you choose to use digital photography or scan in digital photographs, I hope you have fun exploring the options.