Semi-automatically processing a photo series

Digital Shoebox

GIMP is not the most ideal program when faced with the task of editing hundreds of JPEG format photos. This is where the classic command line programs stand out.

BY TILL KAMPPETER

A yield of only five quality photos from an entire roll of film may seem frustrating, but is not bad for an amateur. It is no accident that professional photographers shoot a great deal of film material. This yield rate is only higher with digital snapshots because the bad photos can be deleted immediately from the camera.

However, there is an upside to digital photo processing. After the the initial selection of photos has been transferred to the hard disk, operation "delete photos" can enter phase two. You can then go on to organize your selected photo data in a sensible manner on your hard disk, for example according to date and time stamps, unless of course the camera happened to be using the wrong time zone setting. Photos in which the camera was held vertically have to be rotated, and **dead pixels** require retouching. Although this is enough work for many a rainy weekend, there is a quicker and more efficient way.

GIMP could be used for much of this work, were it not for several serious drawbacks. The program is not capable of working serially, so you can only work on one photo at a time. Being cumbersome, it is also not suited to presenting a complete view of the photo that would allow you to judge whether or not the result is satisfactory.

You lose digital information each time you handle, load, and save an image with GIMP. Photos, especially those from a digital camera, are usually stored in **JPEG** format. Each JPEG-compression, even in case of re-compressing a JPEG photo you opened previously will affect the quality. This becomes obvious when looking at the size of a file saved by GIMP, which will be considerably smaller than the original file size.

Digital cameras not only store the photo data in JPEG files, but also record information such as the time taken, the exposure time, and other information in an **EXIF**-header. When processing with GIMP, or other image processing programs, this information is lost when the photo is saved.

This is reason enough to look for an alternative image processing program. This article shows how you can work with different image processing tools that are not necessarily standard equipment for all Linux distributions and therefore have to be installed manually. See Box 1. We will be looking into "Digital Image Printing" in another issue.

GLOSSARY

Dead pixels: *Unfortunately, a number of cells in the image sensor of a digital camera may be defective and thus deliver a constant value, e.g. a light green point. The degree to which these dead pixels affect a digital photo varies according to where it is located in the photo. An example is shown in Figure 6.*

JPEG: *Abbreviation for "Joint Photographic Experts Group": defines the standards for digital photographic compression as laid down by this group. Unlike the loss-free standard compression technique used with gzip or bzip2, any information that cannot be discerned by the human eye is excluded. Therefore a much higher compression rate can be achieved at the cost of some information loss. Repeatedly opening and saving this data will eventually lead to a total loss of all information, and thus the photo. JPEG is the standard photo format for digital cameras as well as photo presentations on the WWW. It is not suited to compressing drawings or screen shots, where it produces ugly side effects.*

EXIF: *The "Exchangeable Image File Format" is the standard format for storing photo information in digital images. Digital cameras store manufacturer, model, time and date taken, exposure time, focal length, flash settings, aperture and so forth for every photo taken in this header.*

Sneak Preview

The first task is to view the pictures on screen. A **thumbnail** view will provide the best initial overview. With KDE's Konqueror 3.x this function is located in the menu under *View/Preview*(see Figure 1).

Depending on your KDE version, you can activate the *Preview* item, or choose the data type that Konqueror should display when previewing – in the case of photos, this will be *Images* as shown in Figure 1. The data symbols will gradually change one after the other to thumbnail images. If that doesn't work, then check that *View/View Mode Icon View* or *Multicolumn View* is set.

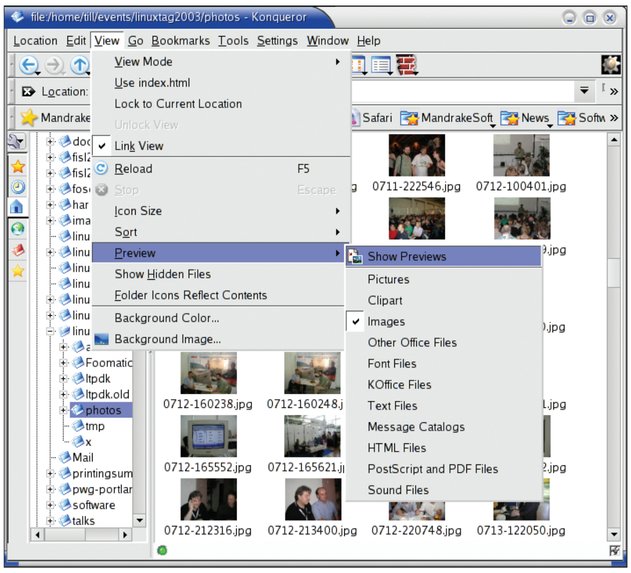


Figure 1: Konqueror with Thumbnail View

Thumbnails are too small to really judge the picture quality; here you will need a full-screen display or an original-size view, where a single pixel on screen represents a single photo pixel. A simple trick allows you to do this with Konqueror: First double click on the thumbnail, then scroll through the image, which will be displayed at its original size, and return to the thumbnails by clicking the *Back* button.

Special programs for image viewing and elementary editing, such as *gqview* [1] and *flphoto* [2], make things even easier. Both show the selected images at full scale next to a list of thumbnails, and this allows you to zoom in and out. With *gqview* (see Figure 2), the thumbnails have to be activated by clicking the (*Index preview*) icon in the tool bar on the left.

gqview is your best bet if you want to sort images in a directory, or delete unwanted photos. A right click on the picture, or on an index entry to the left, will call up a shortcut menu with the available functions. Although the program has no image editing or retouching tools of its own, you can bind user-definable programs to the *Edit* sub-menu of the **shortcut menu** for this purpose. Box 2 shows exactly how that works.

Box 1: Installing picture processing software

Unless otherwise specified, all of the following programs can be installed as RPM packages on SuSE, Red Hat and others using

```
rpm -Uvh package name.rpm
```

If the package manager reports any missing packages required to complete the installation, you will need to add these before attempting to install again. Mandrake Linux users can save themselves the extra mileage by typing

```
urpmi package name.rpm
```

instead of *rpm*. Keep your installation CDs handy! If the Red Hat packages do not run under Mandrake, you can remove them by typing *rpm -e packagename*.

If you want to compile any source code available in the packages, you will need the *Development* package groups, among others, for your distribution's package installation program. Do not forget to install the *devel* packages with the libraries needed by the programs you will be compiling. After doing so, the following commands should do the trick

```
tar -xvzf name.tar.gz
cd name
./configure
make
su -
root password
make install
```

The *README* and/or *INSTALL* files of the packages will inform you of any changes to this routine, as in the following example.

gqview [1] is supplied with many distributions; alternatives, both source code as well as RPMs for Red Hat 7.x, 8 and 9 are offered as downloads on the project website, although only the unstable 1.3.x version of the latter is available. The installation also requires at least GTK+ 1.2.x and *gdk-pixbuf* 0.10.0 for *gqview* 1.2.x, or GTK+ 2.2 for *gqview* 1.3.x as the case may be.

flphoto [2] has been bundled with Mandrake Linux since version 9.1. The project site offers not only the source code, but also a RPM file for Red Hat Linux. *libfltk* 1.x (*http://www.fltk.org/*) must be installed, but full functionality is not available until you install *libgphoto2*, *libcups* and *libexif* (*http://libexif.sf.net/*), which also requires *-devel* packages for compilation.

jhead [4] 2.0 will be available with Mandrake 9.2 (*http://www.mandrakelinux.com/en/cookerdevel.php3*). Mandrake 9.1 has an older version, while SuSE and Red Hat do not offer it at all. The *Jhead* website offers the source code, RPMs for Red Hat, and binary executables for different operating systems. You only have to copy the latter to */usr/local/bin/* and set execute rights for all users using

```
chmod 755 jhead
```

This is also necessary after the *make* step if you are compiling yourself (in this case there is no *make install*); you have to copy the manpage to the required location using

```
cp jhead.1.gz /usr/share/➘
man/man1
chmod 644 /usr/share/man/man1/➘
jhead.1.gz
```

The *libjpeg* library [5] is part of most distributions, and *jpegtran* should also be available everywhere. If not, you may need to install an accessory package from your installation CDs (such as *libjpeg-progs* with Mandrake Linux). *ImageMagick* is usually included with most distributions. In case of doubt, the project page will point the way to source code, RPMs for all Red Hat versions (in the *linux* subdirectories of the mirror sites) and binary *.tar.gz* packages (in the *binaries* subdirectories). The installation of binary packages is described in the *QuickStart.txt* files on the download server.

Only Debian, and Mandrake 9.2 in future, supply their users with *jpegpixi*. In order to compile the source code, you need both the *libjpeg* library and the associated *-devel* package.

Preparing Your Presentation

flphoto (see Figure 4) cannot move a photo around on the hard disk, but presents an integrated photo processing environment. If necessary, photos can be loaded directly from the digital camera using the *Album/Import/Camera* menu item. The tool is based on GPhoto2.

flphoto assembles your images in lists, so-called "albums", adds image files or directories using *Album/Import*, deletes images from an album (not from the hard disk) and allows you to reorganize your photos by drag and drop. The most common image editing task like rotating, cutting, scaling (all in the menu *Image/Transform*), brightness/contrast and even the removal of redeye in flash photography (in the *Image/Touch-up* menu) are easily done. When rotating or cutting, the program does not re-compress JPEG images and thus avoids quality loss.

flphoto also supports all kinds of presentations: *Album/Slideshow* starts a slideshow, *Album/Export* exports all the images in an album to the Web, and the excellent printer functions (no wonder, *flphoto* is by *Michael Sweet*, the author of the **CUPS** printing system) provide glossy prints.

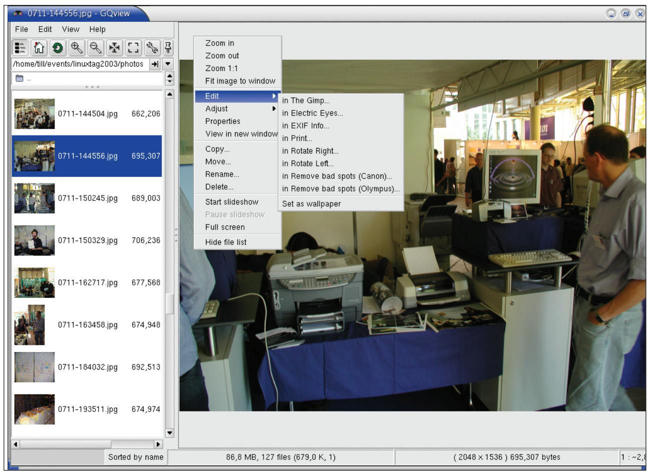
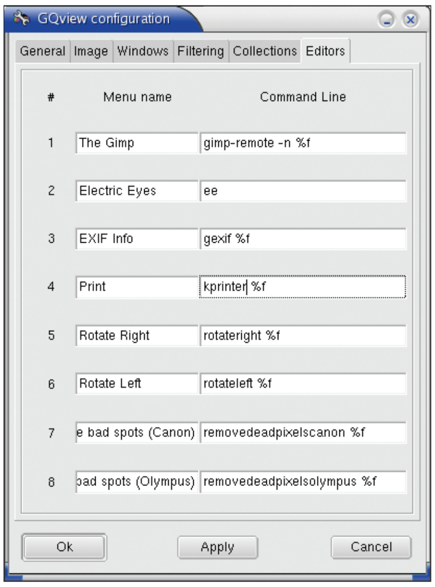


Figure 2: The shortcut menu for the picture viewer in the gqview main window offers a full range of tools for organizing photos on your hard disk, amongst others in the configurable "Edit" menu described in Box 2

No Contradiction: Command line Image Processing

Usually, with image editing programs, the choice of tools will depend on what you can see. This works fine with just a few images; but if you need to edit hundreds of files, it is preferable to use the command line. What is even more amazing is the fact that some tasks can only be done with command line tools.

A classical application for such filtering tools is the rotation of photos taken by cameras held vertically. Even if the camera has an orientation sensor, it will save your photos in landscape format with the subject matter lying on its side. You should rotate all of these photos before presenting your photo collection, so that they are properly oriented.



#	Menu name	Command Line
1	The Gimp	gimp-remote -n %f
2	Electric Eyes	ee
3	EXIF Info	gexif %f
4	Print	kprinter %f
5	Rotate Right	rotateright %f
6	Rotate Left	rotateleft %f
7	e bad spots (Canon)	removedeadpixelscanon %f
8	pad spots (Olympus)	removedeadpixelsolympus %f

Figure 3: The image editing tools defined here populate the Editing menu shown in Figure 2

jhead [4] and the *jpegtran* tool from the JPEG library package, *libjpeg* [5], are recommended for this task. *jhead* parses the EXIF-Header; *jpegtran* rotates without compression, and therefore without loss of quality.

If your digital camera is equipped with an orientation sensor (and you had it on when photographing, see Figure 5), the following command will align all your photos in their correct orientation, provided you are using *jhead* version 2 or higher:

```
jhead -autorot *.jpg
```

jhead reads the orientation information stored by the camera in the EXIF header and, with the help of *jpegtran*, rotates the images. Unfortunately, this tool does not work for all possible camera positions. It fails when the camera is held on its head (trigger pointed down), or the lens is pointed straight up or down.

Manual rotation is possible with older *jhead* versions.

```
jpegtran -rot 270 original↩
photo.jpg > rotatedphoto.jpg
```

will do the job, but *jpegtran* will not transfer the EXIF header from *originalphoto.jpg* to the target file *rotatedphoto.jpg*; furthermore the source and target files must be unique. Both these problems can be solved using the following command

```
jhead -cmd "jpegtran -rot 270 ↩
&i > &o" pic.jpg
```

jhead can use every conceivable image editing command with the *cmd* option; *jhead* saves the EXIF data from *pic.jpg*, carries out the given command (where it

GLOSSARY

Thumbnails: *Scaled down images that make it possible to view multiple of images on the screen at the same time. Many file managers, as well as special image viewers and Web photo albums, support this feature. The on-screen displays of many digital cameras provide a thumbnail viewer, which is available in replay mode when you move the zoom control toward a wide angle setting.*

Shortcut Menu: *When you right click an item, a menu with various options or tools is displayed.*

CUPS: *The "Common Unix Printing System"[3] has now become the standard printing system for most Linux distributions. It replaces, the older LPD ("Line Printer Daemon") as it offers various advantages when using modern printers, or printing in networks. CUPS is extremely easy to configure, as well as giving users access to the full range of printer or driver functions.*

Box 2: Image editing with *gqview*

gqview allows you to specify external image editing tools for via the *Edit* sub-menu of the shortcut menu for an image. These will typically be programs like The GIMP, Electric Eyes, and so on. Tools can be added using *Edit/Settings..* in the *gqview* main menu. In the dialog box that then appears, click on the *Editing Program* tab. Here you will find eight pairs of lines for input; the lines on the right are for command line input, while the ones on the left are for the menu entries. The place holder *%f* in the command line represents the name of the file to be edited. The programs listed here don't have to be graphical applications. Also, non-interactive command line programs can be used, such as the script files from this article: *rotateleft %f*, *rotateright %f*, *removedeadpixels %f*. Figure 3 shows an example.

uses *&i* for the output file and *&o* as the name of the temporary file), inserts the EXIF data in the latter and replaces the output file with the temporary file. So it is possible to use arbitrary image editing commands while still retaining the EXIF information, particularly when manipulating images using *convert* from the ImageMagick package [6].

jhead -cmd "..." is especially useful when the command itself can only edit an explicitly named file and you need to process a large batch of image files. We can ensure that all JPEG images in the working directory will be rotated one after the other by using the following command:

```
jhead -cmd "jpegtran -rot 270 ↩
&i > &o" *.jpg
```

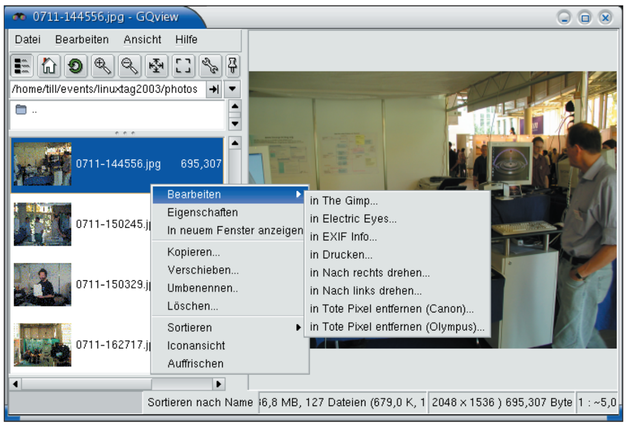


Figure 4: flphoto may not be able to move images around on the hard disk, but can do practically everything else

With *jpegtran* the rotation angle is limited to steps of 90 degrees.

- *-rot 90* rotates 90 degrees to the right,
- *-rot 270* rotates 90 degrees to the left,
- *-rot 180*: rotates through 180 degrees.

There is another drawback you should be aware of. Although *jhead* protects the EXIF during image manipulation, the thumbnail will still be on its side, and its horizontal resolution will exceed the vertical.

To avoid excess typing, you might like to create two small **shell scripts**: *rotateleft* is as follows:

```
#!/bin/sh
jhead -cmd "jpegtran -rot 270 ↩
&i > &o" $*
```

The second script with the name *rotateright* is a follows:

```
#!/bin/sh
jhead -cmd "jpegtran -rot 90 &i↩
 > &o" $*
```

Make both programs executable using

```
chmod 755 rotateleft rotateright
```

and move them to */usr/local/bin*. Provided this directory is in the search path, you can easily rotate your images to the left, using *rotateleft pic1.jpg*, or right *rotateright pic2.jpg*.

Removing Dead Pixels

Batch image editing via the command line is also good for removing dead pixels: As long as you keep the same camera, the faulty sensors responsible for dead pixels will be the same on all photos (see Figure 6).

jpegpixi [7] is used for retouching. *jhead* is again responsible for automating the image correction process for the series. In order to locate the position of the dead pixel, you will need a program that not only displays the pixels, but also gives their co-ordinates when passed over by the mouse. Both GIMP

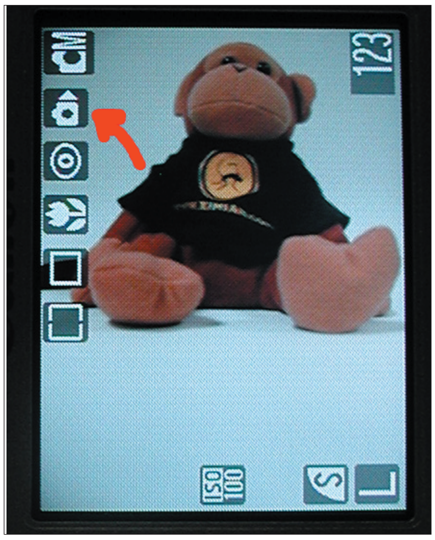


Figure 5: The Canon Digital IXUS 400 automatically recognizes whether the camera was held horizontally or vertically – as shown (marked with the arrow) by the orientation symbol on your screen – and stores all the photos in landscape format, while recording the correct orientation in the EXIF header

and *display* from the ImageMagick package are capable of this. It is easiest with GIMP.

jpegpixi simply uses interpolation to replace an area defined in the command line. Of course this will work with other disturbing elements, and not only dead pixels. The tool re-compresses the image data block and in doing so erases a part of the image. As the other blocks remain untouched, the image quality is maintained to a greater extent.

First, you must locate the faulty pixels in your camera's sensor. The best way to do this is to take a pitch black photo, that you can study on your screen. To do this, hold something black in front of the camera lens (with some cameras it is easiest to put on the protective cap). Underexpose the photo by closing the aperture to the smallest setting and using the shortest shutter time.

For cameras with no manual settings, set the lighting correction factor to the lowest value. But in any case make sure the flash is turned off, so that no light creeps in between the lens and the black object.

As the image should be a uniform black, it is not necessary to focus the camera. Additionally, set the resolution to the same setting as the photo that is to be corrected.

Load the black photo in GIMP, and zoom in until a pixel in the photo is represented by a single pixel on screen (*View/Zoom/1:1* in the shortcut menu of the photo). An alternative method is zooming with the keys [=] and [-].

Now search the photo for light colored pixels on the black background! If you don't find any, then you can boast that your camera's sensors are a 100%, and you will not need to do any retouching. However, if you do find a dead pixel, zoom in on the area, e.g. zoom to *4:1*, and move the mouse to the upper left hand corner of an imaginary square that covers the area where the dead pixel is located. Then move the mouse to the lower right hand corner of this area, and note the co-ordinates. Read the co-ordinates on the lower left hand bar of the photo. If the measurement is in cm or inches, drop down the context menu, and click on *View/Dot for Dot*, to get the pixel mode.

Remove the dead pixel using the co-ordinates of the upper left corner and the side length of the square on the photo, *originalphoto.jpg*, with the following:

```
jpegpixi originalphoto.jpg ↩
correctedphoto.jpg 350,1609,8↩
 1585,865,8
```

We were unlucky and discovered two dead pixels when we tested this camera. The first of these was found at co-ordinates (350,1609) for the upper left hand corner and (358,1617) for the lower right hand corner, and a second dead pixel was located in an area defined by the co-ordinates (1585, 865) and (1593, 873).

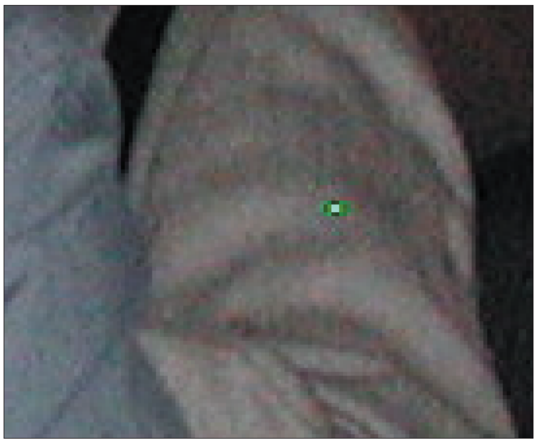


Figure 6: A dead pixel (the light green point in the middle) in an enlarged portion of the image. The fact that it looks blurred is down to JPEG compression

Both areas had a side length of eight pixels. The corrected photo was stored in *correctedphoto.jpg*. Put the black photo through this treatment. The new, corrected photo should be completely black.

A series of photos can be corrected automatically using the following *removedeadpixels* script:

```
#!/bin/sh
jhead -cmd "jpegpixi &i &o 350,↩
1609,8 1585,865,8" $*
```

Afterwards, you can move the script to *usr/local/bin* and make it executable. Now you can correct all of the dead pixels using the following command

```
removedeadpixels *.jpg
```

Timestamp Correction

If you not only want to leave your snapshots to posterity, but also to classify them chronologically, the EXIF header that stores date and timestamps, amongst other things, is a good thing. Unfortunately, the camera is only as accurate as the settings the user supplies. Forgetting to adjust the settings when travelling in different time zones (or even forgetting to change between summer and winter time) produces photos with an inaccurate timestamp.

Luckily, it is possible to correct this mistake later without too much difficulty – again using *jhead*.

Let's say you took some photos in Rio de Janeiro, Brazil, but forgot to change the time on your camera from Middle European Time to the time zone for Brazil. All of your photos will have the time stamp that is five hours ahead of the time they were actually taken. You can use the following command to turn back time for the timestamp:

```
jhead -ta-5 *.jpg
```

The *-ta* flag contains the vector by which the timestamp should be adjusted. You can enter full hours, or alternatively hours and minutes (*-ta+3:30*). To adjust for a day (if you got the camera date setting wrong), use a 24 hour vector.

If you want to assign the last file modification date (as shown by *ls -l*) as the timestamp, you should use this command:

```
jhead -ft *.jpg
```

It is even possible to replace the non-intuitive names assigned by the camera with names based on the exposure date and timestamp:

```
jhead -n *.jpg
```

In this case *jhead* will replace only typical camera-assigned names (i.e. names that use lots of numbers). Any photos that have already been renamed will not be touched. To change all names, use:

```
jhead -nf *.jpg
```

jhead even allows you to assign arbitrary personalized names, and offers more possibilities for manipulating the EXIF data and JPEG files. You can read more in the manpage *man jhead*. ■

GLOSSARY

Shell script: *Programs that use the same commands as are run in the command line (for example a terminal window). The name "shell script" derives from the system program that runs the shell script (and command line instructions), the so-called "shell". The first line of a shell script (#!/bin/sh) specifies the shell the script should be run by, in this case the standard shell /bin/sh.*

$*: *When executing a line of shell script, the shell replaces the string in the line with command line parameters passed to the script when it was called; in this case with the name of the image to be processed.*

INFO

[1] GQview: *http://gqview.sourceforge.net/*

[2] flPhoto: *http://www.easysw.com/~mike/flphoto/*

[3] CUPS: *http://www.cups.org/*

[4] Jhead: *http://www.sentex.net/~mwandel/jhead/*

[5] Independent JPEG Group: *http://www.ijg.org/*

[6] ImageMagick: *http://www.imagemagick.org/*

[7] Jpegpixi: *http://jpegpixi.sourceforge.net/*