

Photomosaics with Metapixel

TILE BY TILE

If you're looking for more than a convention tile design, try Metapixel, a free tool that creates attractive photomosaics from a collection of digital images. **BY OLIVER FROMMEL**

You may be familiar with large-scale posters that use smaller photos to create a complete image [1]. This type of image is called a photomosaic. A photomosaic uses small photos in the way that a conventional graphic image uses pixels. You need to view the photomosaic image from a distance to see the overall effect.

Of course you could put a photomosaic together manually, but the process would be extremely difficult and time consuming. Linux users can now turn to a simple command line program to take care of this chore. Metapixel [2] by Mark Probst evaluates a collection of digital images and takes care of the photo montage.

Preparing Your Tools

The program is easily set up and only requires the jpeg and PNG libraries,

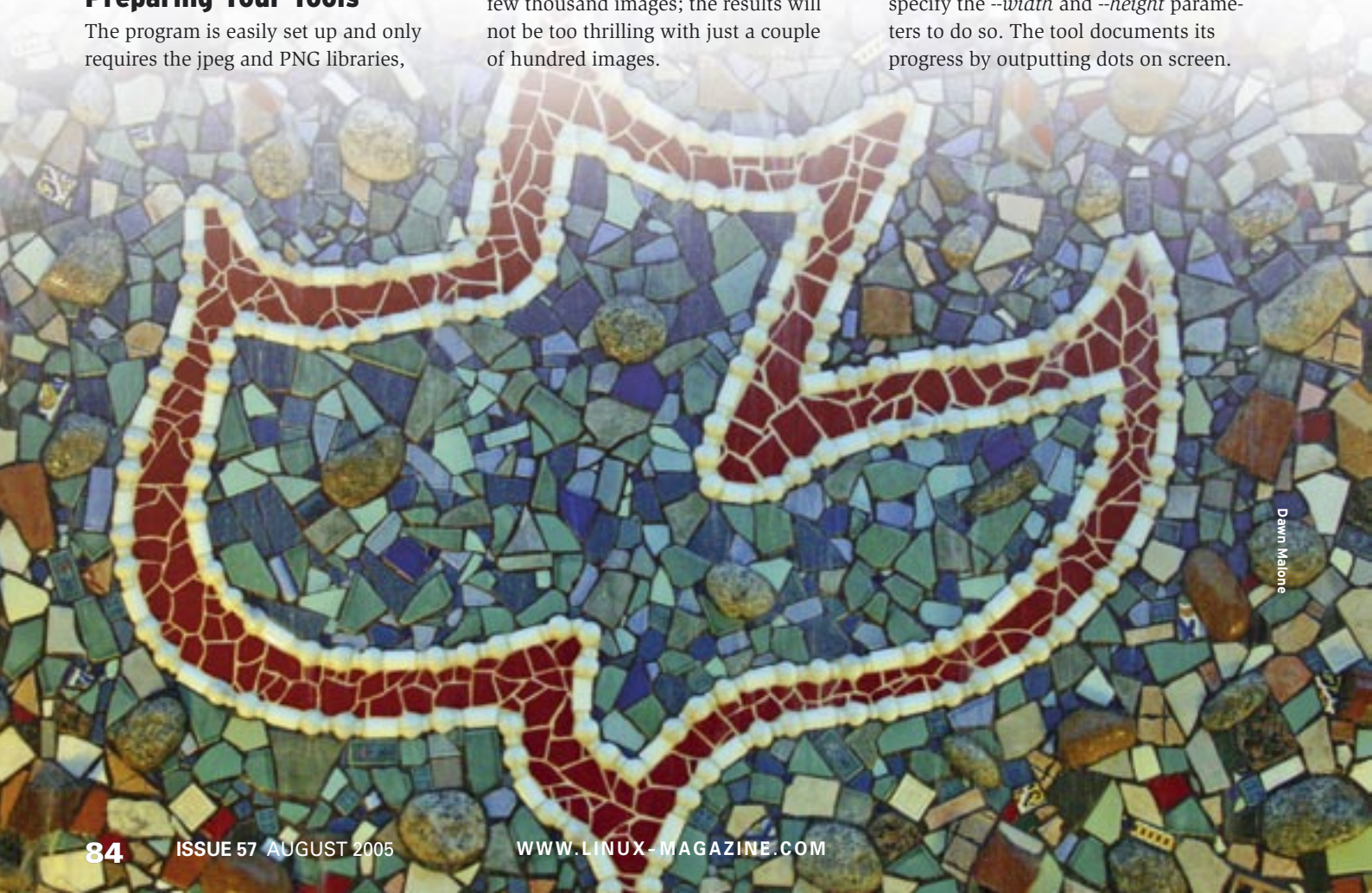
along with matching developer packages. The script included with the package that helps you prepare your digital image collection additionally requires Perl; but most normal Linux systems have Perl installed anyway. After unpacking, first run *make* then type *su -c make install* to put both the executables and the manpage in the appropriate system directories. If you need to change the install path, you can edit the *Makefile* to set a *PREFIX* before performing these steps.

You need a fairly large collection of digital images to make up an attractive mosaic. If your collection is too small, Metapixel will not have enough raw material to reflect the colors and edges in the original. In fact you will need a few thousand images; the results will not be too thrilling with just a couple of hundred images.

Your collection will probably contain photos of different sizes, and you will need to scale these images, as it is difficult to make a mosaic if the tiles are not the same size. The *metapixel-prepare* Perl script takes care of this. The script expects you to pass it the directory with your photo collection and a directory where it can store the scaled images:

```
metapixel-prepare z
imagecollection z
mosaicpics
```

The *-r* (for recursive) option tells the script to search subdirectories. If not told to do otherwise, the script will scale your pics to 120x120 pixels. If you prefer to use a different image size, you can specify the *--width* and *--height* parameters to do so. The tool documents its progress by outputting dots on screen.



Dawn Malone

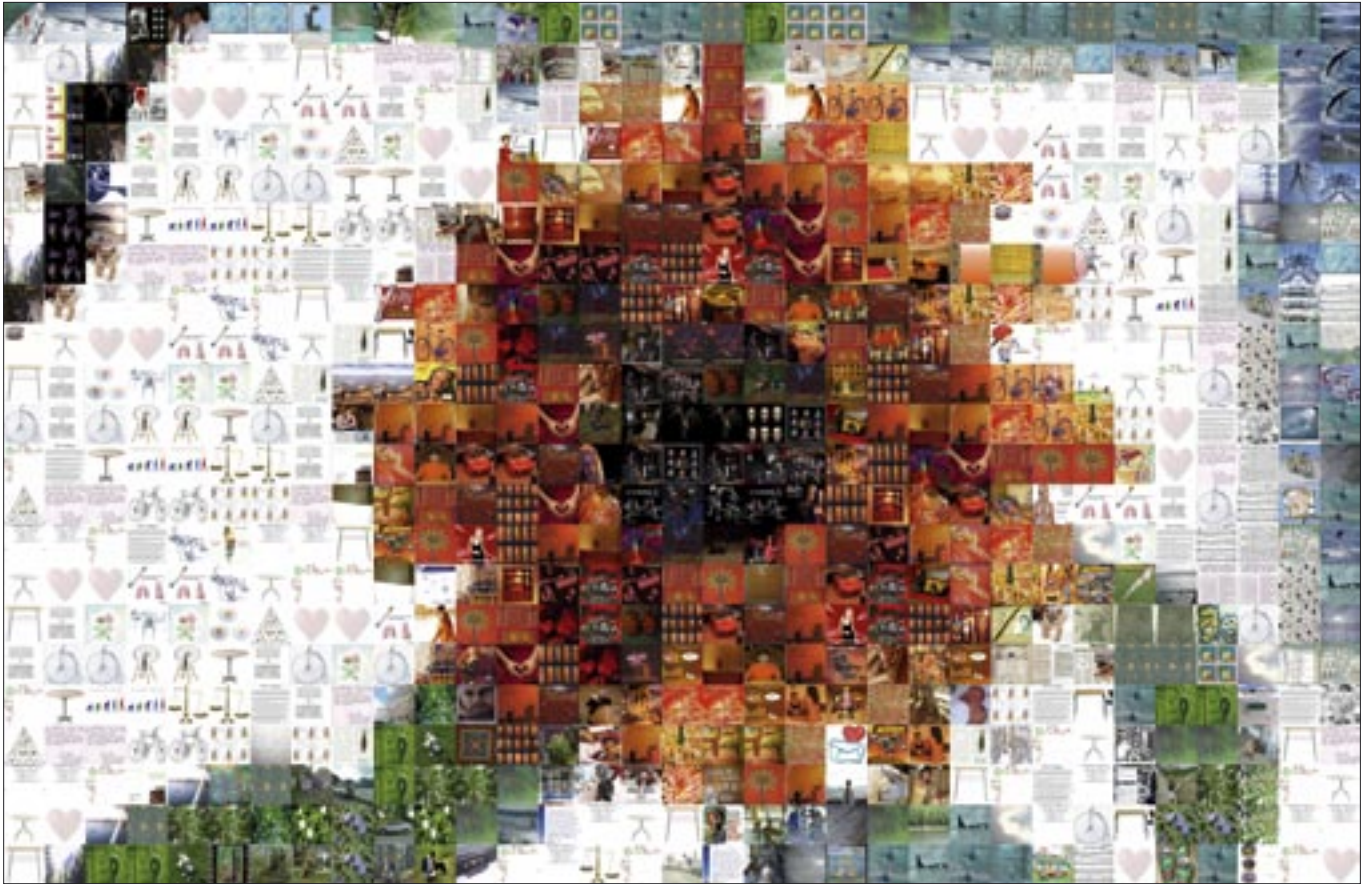


Figure 1: The simplest type of photomosaic, using a picture of a sunflower as a template.



Figure 2: An enhanced version of the photomosaic using more image tiles, and thus showing more of the detail contained in the original template file.



Figure 3: The image tiles in a collage overlap.

Note that the script not only scales your images, but analyzes them at the same time, ascertaining the color and contrast values to help Metapixel put the mosaic together later. The script stores this data in the target directory in a file called *tables.mxt*. If you are interested, you can take a look at the file; it is a text file in a strange kind of Lisp format. If the prepare script crashes or encounters some kind of problem, you might like to set the `--debug` switch to find out more about the issue.

Lots of Dots

After preparing your image collection, you can move on to creating the mosaic. In its simplest form, this involves specifying a template file, the name of the output file, and the directory with the image collection. The following example takes an image called *flower.jpg* and creates a mosaic called *flower-mosaic.png*, using the image collection in *mpix* to do so.

```
metapixel 2
--metapixel flower.jpg 2
flower-mosaic.png 2
--library mpix
```

The `--metapixel` switch tells the program to create a photomosaic without overlapping images. Figure 1 shows the results for this command.

You might not be entirely convinced by the results, as the resolution is fairly coarse. To make a more attractive photo-

mosaic, you need to set the size relationship between the template and the individual image files. If the template is fairly small, meaning that you can't use too many mosaic tiles, the motif will be difficult to recognize. On the other hand, if the tile images are too small, you lose the mosaic effect. Although the template motif is easily recognized, you will probably not notice the individual images in the tiles.

The dimensions have to be set to reflect the finished product you are aiming for. If you are printing a mosaic, you should aim for a larger scale format. And this dictates the size of the image on your computer. For example, if you are aiming for a DIN A3 formatted mosaic (and this is actually quite small), the mosaic will need to measure at least 4000 by 6000 pixels. You may need to scale the template up in Gimp. In this case 33 tiles will fit into the mosaic in the horizontal direction. Figure 2 shows a more attractive mosaic based on a bigger template file.

Overlapping Photos

If you are unable to achieve the kind of results you are looking for, despite having enough tile images and despite having set the size of the template file, Metapixel still has a few tricks up its sleeve. For example it can superimpose the tiles over the original image to make the image more clear, even if the tile images don't fit properly. This option is appropriately named `--cheat`.

Alternatively, you can choose a different kind of mosaic, which Metapixel refers to as a collage. This variant allows the tiled images to overlap. To create a collage, set the `--collage` option when running the program:

```
metapixel 2
--collage flower.jpg 2
flower-collage.png 2
--library mpix
```

The results of this operation are shown in Figure 3.

After experimenting for a while, you may find just the right settings to give you that perfect mosaic; to preserve these settings, you can store them in the `~/.metapixelrc` file. For example, you can set the directory containing your image collection using the *library-directory* keyword followed by the directory name in quotes. Settings need to be surrounded by parentheses. The source distribution includes a sample configuration file.

Beware of Patents

Assuming a large image archive, Metapixel can give you attractive photomosaics with very little effort, but you should be aware of the patent issues surrounding photomosaics [3]. Photomosaics are covered by several patents controlled by the company Runaway Technologies [1]. The details and enforcement of these patents may vary depending on the interpretation and jurisdiction. At a minimum, you will not be able to exploit the results of Metapixel commercially. See [4] for more information on software patents. ■

INFO

- [1] The term "photomosaic" is a registered trademark of Runaway Technologies, Inc. The company has also patented its own photomosaic software and claims a patent on the photomosaic "look and feel." <http://www.photomosaic.com/>
- [2] Metapixel: <http://www.complang.tuwien.ac.at/~schani/metapixel>
- [3] Photomosaic patents: <http://www.photomosaic.com/rt/patent-index.html>
- [4] More about software patents: <http://swpat.ffii.org>