Optical Character Recognition (OCR)

By Andy Pepperdine

Optical Character Recognition is the process whereby a picture of text is turned into a textual form that can be edited by a computer. It is a difficult job to do what we do so easily with our eyes, and there has been a lot of work done on the subject with mixed results.

This paper is intended to describe what is available free of charge. There are some excellent proprietary programs issued with scanners and if you have a system that can run those, it may be the best option for you. This month we will be looking at the cases where that option is not available, either because the scanner did not provide a good one, or you are running an operating system that can execute it.

Preliminaries

Before I start, I must ask that you check that you have the relevant copyright protections to enable you to make the copy and use the results as you wish. If in doubt, get advice. I will be using a printed version of the introduction to the Charter of Fundamental Rights of the European Union purely as an example. Since I printed it out, it has moved site to
http://www.europarl.europa.eu/charter/default_en.htm

What will be covered?

The process of OCR starts with a picture of the text, usually from a scanner. This might be as a JPEG picture, or a PDF form of a picture. PDF's that contain text can usually be manipulated by a PDF reader (e.g. Adobe Acrobat) so you can simply cut and paste from them. If they contain a picture of the text, this is not possible and OCR is needed.

The text may not be very clear and/or it may be slightly on a slant. OCR readers are not very good at handling pictures that differ too much from lines of horizontal text. The GIMP will be used to improve the picture to increase the fidelity of the transcription into text by cutting the piece required, straightening it up, and applying a filter to sharpen the contrast.

The next stage is to process it with the OCR engine. The best of the free ones by all accounts on the Net seems to be *tesseract* and this will be used.

There is some progress being made in handling text that curves due to the scanner not being able to see a page of a book completely flat. This aspect will not be covered here, but see the final reference for more information.

Preparation

I'm using Linux, Ubuntu 9.04 Jaunty Jackalope, to prepare the demonstration. Other versions of Linux will be similar.

I will assume that you have somehow obtained a JPEG file with text in it either by scanning it in or via a digital photograph. Linux scanning is handled these days very well by Xsane and this package will need to be installed. If you do scan, then it is best to scan in grey scale if you know that you

will be converting to text afterwards. Also, it is best to use a high number of dots per inch (at least 150, preferably 200) to improve the chances of good conversion. If any rotations have to be applied before sending it to the OCR engine, then even higher densities are required – the higher the better.

To manipulate the image, you will need the GIMP, which is installed by default in almost all distros, and is available for Windows, too.

For further processing, you may need the package *netpbm* which can change formats of images to generate the right sort for various other programs.

To turn the picture into text, you will need the package containing *tesseract*. On Ubuntu it is labelled *tesseract-ocr-eng* to process English text. It can also process Dutch, French, German (both Roman and Fraktur scripts), Italian, Portuguese, Spanish, and Vietnamese by means of special dictionaries to help word recognition; if you want any of these, install the appropriate packages, too.

Scanning on Linux

Scanning a document on Linux used to be a nuisance, but a lot of scanners now just work out of the box, but unfortunately, not all yet. You can check what to buy on this site: http://www.sane-project.org/ under the Supported Devices heading.

The program to use is Xsane which gives a useful graphical interface and helps you to adjust the parameters to give a reasonable picture. If you are scanning to do OCR later, then best to scan in Grey scale as it is unlikely you will do a better job than the scanner itself in converting afterwards. So set the Grey scale, and then hit the Acquire Preview button and wait for it to create a view and mark the boundaries it has guessed are what you need. You can adjust the boundaries with the mouse if you wish, for example, to just do part of the page.

Then set the scan resolution (probably the top slider) to what you want. You will normally leave the other parameters alone as they have been selected by the preview scan and I've found them better than fiddling with them, although your experience and scanner may vary.

Then hit the Scan button and wait. When the viewer window comes up, save the image where you want it in the format you want by setting the suffix appropriately. If you are lucky (like Xsane on Ubuntu) then you can save directly as a TIF format file (suffix .tif).

If you are trying the *gocr* program, then save as a PGM file (suffix .pgm).

Process the image

Tesseract expects only a single column of text, and only text. So the first job is to clean up the image as far as possible.

First, fire up the GIMP and open the picture file. In order to select a good rectangle of text, the first thing to do is to straighten up the text if it is on the slant. Using the Rotate tool in GIMP, adjust the picture to give horizontal lines. Then select the area using the rectangle selection.

Then right click in the image and choose Crop to Selection. This creates a new image with just the selected text. Save this to avoid any later errors in the default GIMP file type of xcf to preserve the data where possible.

The next step is to reduce the grey scale to black and white if at all possible. Here the GIMP has a useful threshold filter. The filter does not work on indexed layers so first ensure the image is in grey

scale or RGB mode by checking in the Image → Mode menu. Then use Tools → Colour Tools → Threshold and slide the pointer along to see what looks best. My experience has been that it not easy to see what is really best unless you look at in high magnification. If you use too low a threshold, then spurious ligatures can be formed and other artefacts introduced which can seriously affect the result.

I have found that if you let the GIMP save as a TIF file, then it is not readable by tesseract for reasons I have not been able to determine. To get round this, always work on JPEGs in GIMP and save as such. Then apply a simple set of commands to convert from JPEG to TIF with the command line:

```
jpegtopnm file.jpg | pnmtotiff > file.tif
```

where `file.jpg` is the name of the file you've just saved, and `file.tif` is the file it will create.

These two programs are found in the package *netpbm*, which contains over 200 useful utilities for conversion and processing images.

IMPORTANT: Tesseract was written in the days when scanners were strictly black & white and the standardised image format coped only with those under the extension .TIF. Later, colours and more depth was added as scanners acquired colour vision and the extension became .TIFF. However, tesseract still accepts only the .TIF extension and refuses to read a file with the longer form.

Convert picture to text

Tesseract does not yet have a useful GUI, so it is back to the command line:

```
tesseract imagefile.tif textfile
```

The input file name is the full name of the file. The output file name is just the stem and tesseract appends the extension .txt to it.

If you are processing a language other than English, use the option -l <lang> and it will then use the proper dictionary.

Alternative OCR program

Another open source OCR processor is *gocr*, which has a preliminary front-end called *gocr.tcl*. To get these in Ubuntu, install the packages **gocr** and **gocr-tk**. This can read .pgm files, and Xsane can save directly in that file type.

In my testing, it gave significantly poorer results than tesseract, and the resolution must be greatly increased to get an adequate interpretation of the text. But given enough dots per inch, the results are acceptable, even if you have to do quite a bit of editing. Be prepared to use more time scanning and more space for saved files.

Other considerations

If you have received a picture from an external source, you are limited to how clean you can make it before converting to text. If however, you are scanning, then the greater the resolution, the better the results will be. But there are some fonts and some types of printing that will not create good results whatever you try. This applies especially to some sans serif fonts that lead to printed ligature forms, i.e. some pairs of letters are linked together. These gave uniformly poor results in my

experimentation.

If you have to rotate the picture first, then you will almost certainly lose accuracy as you will see the change in clarity as you rotate. It would be better if you could either scan at a much greater density, or align on the scanner beforehand.

History of Tesseract

The OCR program tesseract was first developed at the HP laboratories at Filton near Bristol, UK, and Greeley, Colorado, US, and in conjunction with the University of Nevada at Las Vegas in the mid-1980s. But no work was done by HP on the character recognition part of the program after 1995. However, HP then released it as Open Source in 2005, and Google subsequently picked it up for their own use and have been developing it ever since.

Other sources of information

There is a good introduction to OCR at https://help.ubuntu.com/community/OCR and links from there to an article in Linux Journal.

For a site describing how to improve the picture look at http://aldeby.org/blog/index.php/how-to-professionally-scan-and-ocr-with-open-source-tools.html

The latest version can be obtained at http://code.google.com/p/tesseract-ocr/ which also has a number of links to other aspects of the work.

If you are a member of IEEE (I am not), then you can access a paper from them which describes how to improve text taken by a digital camera from a book where they apparently say what to do when the lines appear curved, etc. If anyone can review it they would do us a service. You can find it at http://ieeexplore.ieee.org/Xplore/login.jsp?url=http%3A%2F%2Fieeexplore.ieee.org%2Fiel5%2F4283407%2F4283408%2F04283429.pdf%3Farnumber%3D4283429&authDecision=-203