Project Report
Number Plate Recognition

Ribemont Francois

Supervisor: Nigel Whyte

April 17, 2012

Contents

1 Introduction

In this document, you will find a short description of the project. Then, you'll find a section about what I changed from the first design documents. After that, there will be a section about what I learned technically and personally. In the next section, you'll find my Review of the project, and finally in the last section, my acknowledgements to the people who helped me.

2 Description of Submitted Project

Number Plate Recognition (NPR) is a software that reads Irish car plate numbers from pictures. The input of the program is an image, and the output of the program is a text string containing the car plate number. By executing a series of algorithms, the program will be able to take an image as an input and generate the number plate in a text format.

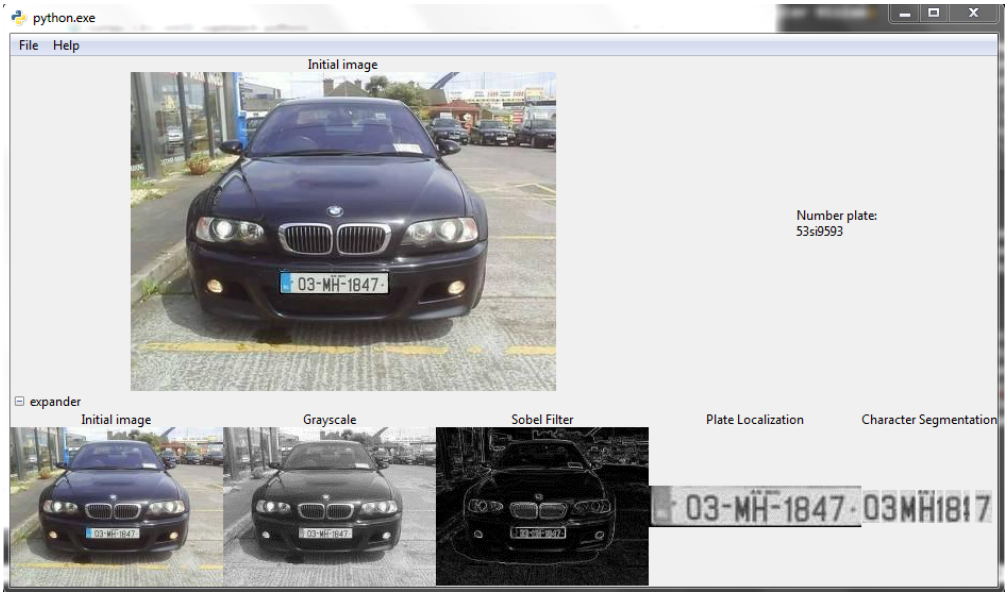

Figure 1: Main window of NPR

3 Description of Conformance to Specification and Design

Most of the specifications and design have been respect, except for a few points in the Design Documents. It was much easier to use function rather than class and objects for the algorithms (no need to instantiate any object for example). I also created a few classes which were really useful on so many aspects. For example, I have created a class Line for the plate localization. Instead of having lists of points, there is a class that gives all the information we need. So it makes the code looks better, it's easier to read, and easier to write. I also created a class Rectangle, and another which is called Progression. The Progress Bar uses it to read information (how much has been achieved when loading), and it also executes the algorithms. It has been made to communicate between the outputs of the algorithms and the thread of the Graphical User Interface (GUI).

4 Description of Learning

4.1 Personal Learning

Computer Vision

During this project, I learned a lot about Computer Vision which I didn't know before this project. It is a very interesting area, where there is always more to learn. I learned how to perform a couple of algorithms such as the grayscale algorithm, or the sobel filter algorithm. It is really hard to get how it works first, but when we persist in it, after looking at simple examples and after trying to make our own ones, it becomes clearer. Trying to split a function when an issue comes up really helps to understand what's going on.

Management

This project was the biggest project I have ever done. A one-year project to do while we still have lectures to attend, and some other projects to do. I learned to manage my free time in order to do everything in time. And also, it learned how to split the tasks over the time.

4.2 Technical Learning

Usage of Computer Vision libraries

To help me doing that, I used a couple of libraries for the manipulation of the data, but I implemented the algorithms I was talking about earlier myself. I used the libraries Numpy, SciPy and PIL. Numpy has a data type that is very good, it permits to manipulate images as matrices. SciPy uses Numpy in order to run, and I used it for loading, saving and resizing images. Finally, I didn't use PIL myself, but Numpy and SciPy need it as a dependency to run.

Graphical User Interface improvements

I learned a couple of things about PyGTK, even though I knew the basics of it. I didn't know how to manipulate a Progress Bar, or how to create Dialog windows, or the manipulation of EventBox for the images to get events. Since GTK is really used in Linux interfaces, it can be useful in the future.

Better skills in Python

I learned very much on Python this year, even though I've used it for the 2 two years for my personal projects. I didn't know anything about list and dictionary

comprehensions which are really powerful and even if though it doesn't look as good as the rest of Python, it is not really hard to use them, but it can get really confusing. I also learned to use the "special" methods in Python (the ones who start and end with a double underscore, such as __str__). They can be useful to debug the code, or to make it much easier to read.

I also learned how to use the thread module of Python in order to move my progress bar with the progress that has actually been done. Threads are powerful but it can get tricky to use them or to share data between them.

5 Review of Project

5.1 What went right

The GUI is fully working. Everything that needed to be implemented has been done. Most of the algorithms have been implemented, such as the grayscale algorithm, the Sobel filter algorithm, the plate localization, and the character segmentation using vertical and horizontal projections. A bit of template matching has been done too.

5.2 What went wrong

I spent a lot of time at the beginning to understand how Numpy worked. Hence it's has been done, things went much faster.

5.3 What's missing

The final algorithm template matching is not working. I still haven't figured out where the problem is and why it doesn't work. The algorithms are not robust enough to handle every images, such as the plate orientation, if it is not straight, it can't detect the car plate.

5.4 How would I do it differently

If I had to do it again, I would change the programing language for only one reason: the performance. I read that Python is not slow for basic stuffs, but when it comes to computer vision, Python is not the fastest programming language. It is due to a low speed to perform loops. Sometimes loops can be replaced by list/dictionary comprehension, but not in this case. So, with the same program written in C++, it would have been much faster. If I had to give an advice to someone who would like to do this project, I'd say to him/her: "If you want to use this program everyday because you need it, so it means you need good performance, then you should consider using C/C++, or if you want to write this program to learn more about Computer Vision, but you don't need good performance, then you should write it in Python, because in Python, you only care about the algorithms themselves, since you don't need to spend too much time on pointers (in C, for instance)".

6 Acknowledgements

I would like to thank some people who helped me during this project:

- Nigel Whyte: My supervisor who helped me doing my project and was always answering my questions.
- The people of #python-fr on the Internet Relay Chat (IRC) freenode.net server who helped me a lot understanding how Numpy actually worked.
- Paul Barry who took some time to look at my code in order to make it better.
- Numpy's documenters. They wrote a very good tutorial[1] on how to use Numpy.

[1] http://www.scipy.org/Tentative_NumPy_Tutorial

Acronyms

GUI Graphical User Interface. 7

IRC Internet Relay Chat. 8

NPR Number Plate Recognition. 3