RoadEye

by

Marcelo Dias Avelino (0840416)

CMI-Program *Technical Informatics* – Rotterdam University

April 28, 2014

First supervisor *Mrs. T. Ubert*
Second supervisor *Mr. A. van Raamt*

Abstract

ala

Acknowledgements

Wie kan je zoal bedanken? Denk aan de begeleiders en voorbereiders van je afstudeerproject, familieleden en andere personen die je geadviseerd of gemotiveerd hebben. Het is gebruikelijk om dit voorafgaande aan het verslag te doen. Dit bedanken mag ook in de inleiding gebeuren. Bijvoorbeeld: Bij het opstellen van dit verslag heb ik dankbaar gebruik gemaakt van 'metathesis' van *Donald Craig (donald@mun.ca)*.

Contents

Introduction

Background information

This project has been made available by CGI Nederland BV, which is a daughter company of the CGI Group Inc. located in The Netherlands. CGI is a multinational IT (Information Technology) company which provides consulting, systems integration, outsourcing, etc. on a multitude of different technological areas. Because of its affinity with these different kinds of technology, CGI researches various technologies to solve problems facing modern society. As a result of this drive, the idea for this project arose: How can people be given the ability to help track stolen cars, and by extension improve their community? Since everyone nowadays carries a smart phone around in their pocket, there is a lot of untapped processing power which could be used for this purpose. This led to the invitation of a student to research and implement this project as the topic for a bachelor thesis. I chose this project because of its relation with Computer Vision, a computer science topic I have worked with before and find interesting.

Problem

Even nowadays, it is still difficult for law enforcement to follow up on every case of stolen property, e.g. cars, simply because they cannot be everywhere at once. A possible way to improve law enforcement's efficiency would be by including the aid of civilians through the voluntary provision of gathered information from one's environment. This used to be a very difficult issue because required people to carry dedicated, and often large, hardware devices which made the gathering of information possible. Since the breaking of the smart phone age, a large number of people carry a computer in their pockets. This Internet-connected all-purpose device allows for a whole range of new possibilities and with the right software, gives the user the possibility to help his own community and therefore help making it a better place. One of these possibilities is to capture video images using the smart phone's camera. These images when paired with a Computer Vision algorithm are able to produce amazing results. This project will therefore be focused on applying the Computer Vision discipline to solve this problem and try to answer the following question: *How can license plate information be gathered from images of a video camera using software?*

But this only concerns the problem of this project in a more general way, without taking any requirements or conditions into account. Because the software must run on a mobile platform it must take all the limitations of such a platform or the environment where it will operate into consideration, i.e. limited processing power, battery life, unstable images or car distance. To keep these conditions in mind while developing the application, the following questions will be answered: *'How can the software be optimized to work in a correct way from within a mobile device?'* and *'What are the limitations of such an application?'*.

Objective

The objective of this project consists in creating an Android smart phone software application that is able to locate license plates in images gathered from the smart phone's camera. From these images, the application must be capable of reading the alphanumeric text displayed on the plates. This information will then be compared to a list of license plate information, which was fetched from an website beforehand. If there is a match, the application must inform a central application of the said match, along with its position when the image was captured and how reliable the recognition is.

Scope

The scope of this project will encompass the development of the android application which will recognize license plates using the images from the phone's camera, the communication with the web page where the license plate information will be fetched from and the communication with the main application where information over the recognition will be sent to. It will only take into account dutch yellow car license plates and might therefore not work with foreign plates. It will take into account, implementation and design wise, privacy concerns according to the dutch law. The software will only be written and tested for Android version 4.2.2, running on an HTC One X. It might therefore not work on different versions.

Methodology

Research method

The method applied for researching the problems of this project and its possible solutions is called *literature review*. This method consists of researching what has already been published, which might be in the form of scientific or engineering papers, journals, thesis, etc., by accredited scientists, scholars or engineers concerning this assignment's topics. This method is applied for searching for potential algorithms which can be used to solve the problems facing the project. Once a group of the most suitable algorithms has been found, the best one must be chosen and the reasoning for this choice must be explained. When the choice has been made, the algorithm can then be implemented using the chosen development method.

Development method

This project will be developed using the *Iterative Application Development* (IAD) method. This development method works by dividing the project into smaller 'sub-projects', called *cycles*, and incrementing them to past cycles, which will ultimately lead to a complete system. Each cycle consists of three phases, which can be repeated multiple times if necessary, called *iterations*. These iterations are: *definition*, *development* and *deployment*.

During the definition phase the goals, limitations and conditions for the current cycle are examined and described. If a previous cycle has been completed, it will be evaluated during this phase. This phase is intended for thinking towards the completion of the project and to achieve a more clear picture of the system as a whole. After defining the objective for the new cycle, the software will be developed. After finishing, the software is then integrated with the software developed in the previous cycles and becomes therefore part of the general project.

This method of software development brings multiple benefits: The complexity of the project is decreased by breaking down the problem into smaller chunks, which allows for faster and more concrete results and makes it therefore easier to get better feedback or to solve critical bottlenecks by being able to discuss them at the end of each cycle. The project development also becomes more flexible by having the possibility to review the requirements and strategies every cycle.

Each cycle lasts 2 weeks and at the end of each cycle the evaluation of the past cycle and the objective for the coming cycle will be discussed with the organisation's mentor.

Algorithms

Plate localization

When searching for possible algorithms with the functionality to find license plates in an image, two main types came forth from the research: feature detection and edge detection.

The feature detection algorithms works by finding so called *features* in a image, which are used to recognize the first image within a second one. These features are segments of an image which must be uncommon, as to reduce the possibility of retrieving a false positive when applying the algorithm, and also consist of something which can be objectively described to a computer. Because of these requirements, the features extracted from an images are usually corners since corners usually only match themselves when compared to other segments in an image. This opposed to flat surfaces or lines, which may appear multiple times in multiple places in the same image. Because this algorithm focuses on detecting the uniqueness of an image and using those attributes to detect themselves in different images, it is difficult to use feature detection for the recognition of license plates for the reason that every license plate contains unique text. The considerable collection of diverse shapes that exist in the Latin alphabet create false positives which are often detected in random and incorrect locations. One possible approach to use this algorithm would be by creating a feature database of every possible alphanumeric character and then finding the highest concentration of text as a possible location.

The other possible algorithm is mostly based on edge detection. This kind of algorithm works by applying an edge detection filter to a grey scale version of the image where the car is present, e.g. the Sobel Filter [ref here] or Canny Edge Filter [ref here]. This creates a binary image where the edges of every shape present in the image are displayed. Because of the nature of one of the characteristics common to every license plate, which is the presence of text, an area with a high density of edges is created. Although license plates are not the only objects which might have such a property, e.g. a fence, it is the most common one which might be encountered while driving. By applying this filter to find horizontal edges it is possible to find the vertical location of the plate, called a *band*, and then vice-versa to find the horizontal location and by extension the plate itself. Due to little information on the performance of the first algorithm and an healthy amount of information regarding this one, this algorithm was chosen.

Text recognition

'Explanation on training Tesseract'

Application

Design

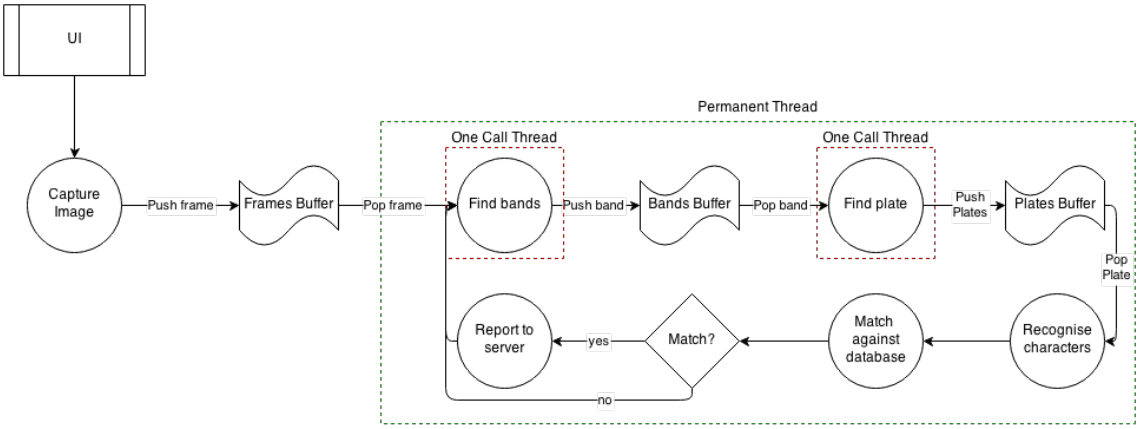

Figure 1: System flow diagram

Implementation

Conclusion and recommendations

Conclusies en aanbevelingen moeten verzameld worden in een apart en herkenbaar deel van het verslag. Hoewel in het hoofdverslag op diverse plaatsen conclusies getrokken kunnen worden, moeten de belangrijkste conclusies samengevoegd en samengevat worden.

Belangrijk is dat het verschil tussen objectief controleerbare conclusies en subjectieve aanbevelingen duidelijk wordt aangegeven. Ook is het aan te bevelen om de belangrijkste conclusies conform de opdrachtomschrijving te formuleren.

References

[1] Lamport L.: *LaTeX: A Document Preparation System*, Addison-Wesley, 1994

[2] Oostrum van P.: *Handleiding LaTeX*, Vakgroep Informatica, Universiteit Utrecht, 1998, http://people.cs.uu.nl/piet/latexhnd.pdf

[3] Wikibooks LaTeX: http://nl.wikibooks.org/wiki/LaTeX

Evaluatie

In de evaluatie reflecteer je over je eigen afstudeerproces. Daarbij moet je vooral letten op de leereffecten. Welke competenties had je nodig? Welke competenties kwam je tekort en moest je zelf verwerven? Waren dit algemene of specifieke competenties? Voldeden de beroepscompetenties aan de standaard van het *HBO-I* (analyseren, adviseren, ontwerpen, realiseren en beheren)? Vielen de algemene competenties in de vijf categorieën van de *Dublin Descriptoren*[1] zoals het verkrijgen van kennis en inzicht, het toepassen van kennis en inzicht, het maken van onderbouwde keuzen (oordeelsvorming), het communiceren (schriftelijk en mondeling) en het verkrijgen van leervaardigheden?

[1] Dublin Descriptoren zijn eisen aan de competenties voor de bachelor en master studies aan universiteiten en hogescholen in Europa.

Appendix A

Achtergrond materiaal

In de bijlagen komen alle gegevens die nodig zijn voor de onderbouwing, maar die de leesbaarheid van het hoofdverslag verlagen.