

A Real Time Vehicle's License Plate Recognition System

Choudhury A. Rahman, Wael Badawy
Department of Electrical and Computer Engineering
University of Calgary
Calgary, Alberta, Canada T2N 1N4
{rahmanc, badawy}@enel.ucalgary.ca

Ahmad Radmanesh
Signals Division
The City of Calgary
Calgary, Alberta, Canada T2P 2M5
aradmanesh@calgary.ca

Abstract

A smart and simple algorithm is presented in this paper for vehicle's license plate recognition system. Based on pattern matching, this algorithm can be applied for real time detection of license plates for collecting data for surveying or for some application specific purposes. The proposed system has been prototyped using C++ and the experimental results have been shown for recognition of Alberta license plates.

1. Introduction

Vehicle's license plate recognition system has been a special area of interest in video surveillance arena for more than a decade or so. With the advent of sophisticated video vehicle detection systems for traffic management applications, number plate recognition system finds wide varieties of places to fit itself beyond just controlling access to a toll collection point or parking lot. It can now be integrated to the video vehicle detection systems which usually are installed in places of interest for intersection control, traffic monitoring etc., to identify vehicle that violates traffic laws or to find stolen vehicles. There are a number of techniques used so far for recognition of number plates such as BAM (Bi-directional Associative Memories) neural network character recognition [1], pattern matching [2] etc. The technique used in the proposed system is based on pattern matching, which is fast and accurate enough for real time applications and is developed for recognition of Alberta license plates with prior knowledge of letters and numbers orientation. Since the orientation and font used for number plates differ in different countries/states/provinces, this algorithm is needed to be modified accordingly keeping its structure intact, if we want to apply this system for recognizing the number plates of those places.

The rest of the paper is organized as follows: section 2 discusses about the system architecture and the steps involved in recognition process, section 3 explains the

proposed algorithm for real time detection of number plates, section 4 shows the experimental results and finally section 5 concludes the paper with acknowledgements and references.

2. System Architecture

The proposed system will be used as a part of intersection surveillance video camera system for traffic analysis. Figure 1 shows a typical intersection of the city of Calgary. Only one license plate is used in Alberta province which is attached to the back side of the vehicle and the camera will be used to track this back license plate.



Figure 1. A typical intersection of the city of Calgary

The proposed system architecture contains three distinct parts: outdoor part, indoor part and communication link. Outdoor part is the cameras installed in different intersections of interest for capturing images. The indoor part is the central control station that receives, stores and analyzes the captured images from all these installed cameras. Communication link can be high speed cable or fiber optic connecting all these cameras to the central control station.

Almost all the algorithms developed so far work by following similar steps. Seven general processing steps have been identified as being common to all number plate recognition algorithms [3]. These are:

Trigger: This may be hardware or software trigger. Hardware trigger is the old approach where inductive loop is used for triggering and this tells when the image should

be captured by detecting the presence of vehicle. Hardware trigger is now-a-days being replaced by software trigger and in operational in many places. In software trigger, image is divided into zones and by image analysis the detection of vehicles is performed.

Image Capture: Hardware or software trigger activates the image capturing device to capture and store image for further analysis.

Vehicle's Presence: This step is needed only if trigger is done after definite time intervals without knowing that a vehicle is present in the captured image. This step compares the captured image with the background image and detects if there is any significant change. If not, the captured image is just ignored, otherwise it moves to the next step.

Finding Plate: This step is for locating the number plate in the captured image. A number of techniques can be used in this step e.g. color detection [4], signature analysis [5], edge detection [6] etc. Any tilt in the captured image is corrected in this step. Once the number plate is located the image is ready for character recognition.

Character segmentation: Segmentation can be done by detecting the transition from dark to light or from light to dark level. Each character present in the number plate produces a band of gray level. So by detecting the similar gray level bands each character can be segmented.

Recognition process: This is the OCR step. The techniques that can be used in this step include pattern matching [2], feature matching [7][8] and neural network classifier [9].

Post process: This is the application specific process step. Depending on the application this step may save the detected number plate for traffic data collection, try to match the number plate to the stolen vehicles database or to open the gate of a parking lot for authorized vehicles etc.

3. The Proposed Algorithm

The proposed algorithm has been implemented for recognition of license plate characters after the processing of captured image and plate detection. Based on pattern matching, the proposed system follows a smart algorithm for recognition of letters and numbers of Alberta license plates. Figure 2 shows a sample of Alberta license plate which contains three letters, three numbers and a dash in between. So with this prior knowledge of character orientation the ambiguity between characters such as: digit '0' and the letter 'O', digit '8' and letter 'B' has been solved.

Moreover, since the first three characters are letters so it is only needed to compare these characters with the

pattern of letters from A-Z only. Similarly, for the last three characters, it is only needed to compare them with the pattern of digits from 0-9.



Figure 2. Alberta license plate

First problem in this character recognition is to identify the area where the characters are printed. This area is usually vertically and horizontally centered. So by taking vertical concentration of colors we can get the top and bottom positions of the characters. Once the top and bottom positions of the characters in the image have been found, the area can be segmented from the image resulting Figure 3. This image is now ready for character segmentation and recognition.



Figure 3. Segmented image containing only characters for further processing

Character segmentation can be carried out by taking horizontal concentration of color. For pattern matching to work effectively, it is needed to find font closely matched with the one used in the number plate. Arial font is used in this case which works reasonably well for character recognition of Alberta license plate. With this font a library is built at first. This library contains the histogram of letters A-Z and digits 0-9. Fifteen different histograms have been generated for each of the characters for building the library. These are: horizontal histograms for (1) full size, (2) lower half, (3) upper half, (4) lower one third, (5) upper one third, (6) lower one fourth, (7) upper one fourth, (8) upper two third of the character and vertical histograms for (9) full size, (10) left half, (11) right half, (12) left one third, (13) right one third, (14) left one fourth and (15) right one fourth of the character.

The flow chart of the process of recognition is given in Figure 4. As shown in this figure, three segments are taken for recognition each time and the library that is used each time depends on which of these 'three segments' sets has been taken. If the 'three segments' set is letter then 'alphabet' library will be used for comparison otherwise 'number' library. There are fifteen different histograms for each of the letters from A-Z in the 'alphabet' library and fifteen different histograms for each of the digits from 0-9 in the 'number' library. The algorithm shown in Figure 4 has to be run twice, taking each of the 'three

segments' set once, for complete recognition of the number plate. i in the flow chart is the iteration operator. s and p are parameters for matching.

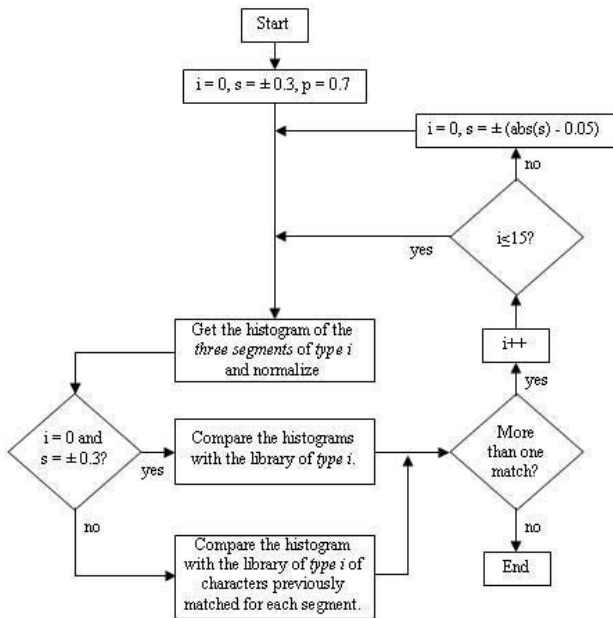


Figure 4. Flow chart for character recognition

The value of i changes for every iteration and this value indicates which one of the histograms in the library should be used for comparison. As shown in the flow chart, the histogram that is extracted (which is governed by the value of i) from segments is first normalized before comparing. Once the normalization process is complete, the segment is ready for pattern matching with the pattern stored in the library. Each matching process gives a set of characters that are closely matched with the segment under inspection. So by doing this process several times with different histogram patterns, the least possible characters are filtered out leaving the correct one.

4. Experimental Results

The proposed system has been prototyped using C++ and tested using a sample of Alberta license plate. Figure 5 shows the process of identifying the number plate character's location by taking vertical color concentration in the image. Scanning horizontally from center to up and from center to down the top (H_1) and bottom (H_2) positions of the characters in the image are found.



Figure 5. Vertical color concentration

Figure 6 shows the character segmentation process. This is done by taking horizontal concentration of color. Since we know that the first three characters are letters and the last three characters are digits, we can easily group them after segmentation for the next step: pattern matching.

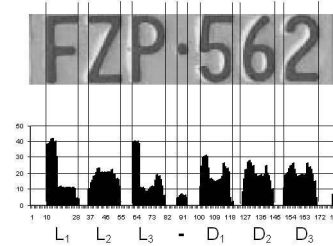


Figure 6. Character segmentation

As shown in the flow chart the fifteen different patterns used in this system are governed by the value of i and are assigned in this way: 0 (horizontal histogram, full size), 1 (vertical histogram, full size), 2 (horizontal histogram, upper half), 3 (vertical histogram, left half), 4 (vertical histogram, right half), 5 (horizontal histogram, lower half), 6 (horizontal histogram, lower one third), 7 (vertical histogram, right one third), 8 (horizontal histogram, upper one third), 9 (vertical histogram, left one third), 10 (horizontal histogram, lower one fourth), 11 (horizontal histogram, upper one fourth), 12 (horizontal histogram, upper two third), 13 (vertical histogram, right one fourth), 14 (vertical histogram, left one fourth). The histogram that is extracted from a segment is first normalized before comparing.

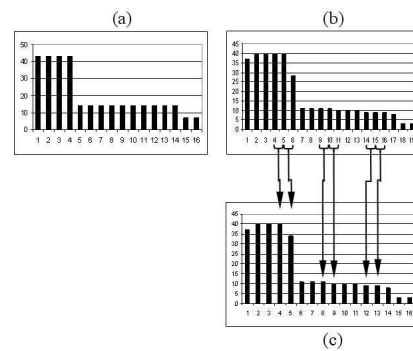


Figure 7. Normalization process

The normalization is done by comparing the width of the segment to the library. For an example, if horizontal histogram is taken for comparison, the maximum width of the three segments in the horizontal direction is compared with the library's maximum. If the width of the segment is greater, the histogram is shrunk evenly in the horizontal direction averaging the value of the histogram of the neighboring positions. Similar process has been used for enlarging, if it is smaller. Figure 7 illustrates

normalization when the width of the segment is greater compared to the library.

Figure 7(a) shows the horizontal full histogram of letter 'F' in the library. Figure 7(b) shows the horizontal histogram of letter 'F' found from the segment. If the width of the histogram of 'F' in the library is the maximum (in this case 16), the histogram width found from the segment (in this case 19) must be reduced to 16 before comparison. This is done and shown in Figure 7(c). Since the width difference is 3, the histogram values of 3 evenly distributed positions in the segment's histogram are discarded, averaging the neighboring values. As shown in Figure 7(c), 5th, 10th and 15th position values are discarded averaging the values of 4th, 6th, 9th, 11th, 14th and 16th positions. 4th position's new value is now the average of 4th and 5th positions previous value. Similarly, 5th position's new value is now the average of 5th and 6th positions previous value and so on.

After normalization, pattern matching process is carried out. This is done by comparing two ratios for each position in the histograms. One is from the segment and the other is from the library. The ratios are the histogram values of each position to the maximum available in that graph. If the difference of these two ratios is within the value, set by parameter s , match count is increased. So by doing this procedure for every horizontal (for horizontal histogram)/vertical (for vertical histogram) position of the segment, we get a match count indicating how closely the segment matches with the character. Iterating this process for every character in the library, we get match count for each character in the library. Now, by assuming the highest match count as 100% match, the algorithm filters those characters of which matches are less than 70% (set by parameter p). So next time when the algorithm takes a different histogram, it compares this segment with only the characters detected previously. If after doing these comparisons for fifteen different histograms, there still exists more than one match, the whole process is repeated with higher sensitivity (s is decreased for increased sensitivity) until a single match is found.

5. Conclusion

In this paper a real time vehicle's license plate recognition system is presented, highlighting some areas of applications where this system can be implemented. The system architecture has been discussed with identifying the steps involved in recognition process. The experiment that has been carried out, clarifies the proposed system as a potential candidate for real time recognition. The experiment shown in this paper assumed ideal weather condition. Research assuming various

weather conditions is in progress. The prototyped system will be integrated to the intersection surveillance video system for traffic surveying or for some application specific purposes discussed in the paper.

6. Acknowledgements

The authors would like to thank the Natural Sciences and Engineering Research Council of Canada (NSERC), University of Calgary and the City of Calgary for supporting this research.

7. References

- [1] Maged M. M. Fahmy, "Automatic number-plate recognition: neural network approach," *Proceedings of Vehicle Navigation and Information Systems Conference*, pp. 99-101, September 1994.
- [2] D. Irecki & D. G. Bailey, "Vehicle registration plate localization and recognition", *Proceedings of the Electronics New Zealand Conference, ENZCon'01*, New Plymouth, New Zealand, September 2001.
- [3] D. G. Bailey, D. Irecki, B. K. Lim and L. Yang, "Test bed for number plate recognition applications," *Proceeding of the IEEE International Workshop on Electronic Design, Test and Applications*, 2002.
- [4] B. H. Cho & S. H. Jung, "Non feature-based vehicle plate recognition system using neural network", *Proceeding of ITC-CSCC 98 International Conference*, Korea, Vol. 2, 1065-1068, July 1998.
- [5] J. Barroso, J. Bulas-Cruz & E. L. Dagless, "Real Time Number Plate Reading", *4th IFAC Workshop on Algorithms and Architectures for Real-time Control*, Portugal, April 1997.
- [6] J. R. Parker & P. Federl, "An approach to licence plate recognition", *Computer Science Technical reports*, University of Calgary, Alberta Canada, Vol. 591-11, October 1996.
- [7] J. Barroso, A. Rafael, E. L. Dagless and J. Bulas-Cruz, "Number plate reading using computer vision", *IEEE Intern. Symposium on Industrial Electronics, ISIE'97*, 1997.
- [8] Shokri Gendy, Clifton L. Smith and Stefan Lachowicz, "Automatic car registration plate recognition using fast Haugh Transform", *Proceedings of the Institute of Electrical and Electronics Engineers*, Carnahan, pp. 209 - 218, October 1997.
- [9] R. van Heerden and C. Nieuwoudt, "Automatic number plate segmentation and recognition", *Department of Electrical and Electronic Engineering Reports*, University of Pretoria, 1998.