A Quadtree Based Vehicles Recognition System

PEDRO BARROSO, JOAQUIM AMARAL, ANDRÉ MORA, JOSÉ MANUEL FONSECA, ADOLFO STEIGER-GARÇÃO
UNINOVA – Instituto de Desenvolvimento de Novas Tecnologias e
UNL – FCT – Departamento de Engenharia Electrotécnica, 2003
PORTUGAL

Abstract – This paper presents a Quadtree based Vehicles Recognition System (SIREVIA), which uses images captured by a digital camera for identifying the vehicle's license plate. A methodology to identify the license plate in the captured image using digital image processing techniques and artificial intelligent techniques is presented. The plate location algorithm is based on the detection of high contrast areas along the image lines. The character recognition is preformed using a quadtree based object detection algorithm and an automatic classifier using decision trees trained for identifying all the alphabet letters and numbers. A prototype of an access control for a car parking is also presented. This application uses a database witch keeps the information about authorized cars, users and different access levels to a specific place.

Key-Words: Licence Plate Recognition; tonal variation; Segmentation; OCR; Quadtree;

1. Introduction

The development of automated applications is becoming more and more a request in modern societies, due to the increasing needs for efficiency and fast decisions. The Vehicles Recognition System described in this paper is one of such applications. The use of image processing techniques and artificial intelligence techniques for fast and automatic recognition of vehicles license plates is a result of nowadays technological development and innovation.

Some common applications for these systems are access control to restricted car parks and zones, traffic and average speed control and tolling [1] [2]. The absence of any devices installed in the vehicle, such as magnetic cards or transponders, and the capacity to work autonomously without the need of the driver's involvement makes the system management very flexible, allowing upgrades in the behavior without the need for major changes in the global system's architecture.

A major example in the area of License Plates' Recognition is Greater London Council's effort to reduce traffic inside the city [1]. This initiative is the most ambitious and sophisticated attempt that was ever done to reduce traffic problems in a city. According to Transport For London [2], 203 computer-linked camera sets have been set up to photograph the license plates of all vehicles entering the city center, with the intent of automatically detecting, recognizing and punishing the infringer vehicles, with a 90% accuracy rate. Since this is a new area of development, quantitative results for comparison are scarce.

In this paper we propose a methodology for automatic detection and recognition of license plates, from static front car images, for controlling the access to restricted car parks. In order to automate this procedure, it is required to correlate computer science techniques and electronics to create a system capable of extracting data from the physical information; a camera for taking the image, a position sensor for detecting the vehicle's presence and processing it using image analysis software.

The proposed methodology focuses on three main steps: the tracking of the license plate in the captured image, detection of objects that are within the plate rectangle and in the color range of the plate's characters, and finally the character recognition. The execution of these steps results in the use of the car plate number in distinct applications.

In early stages of the tracking algorithm conception, some studies were carried through, in order to locate the license plate based on the location of the blue field that delimits European plates. However, the different illumination conditions and plate conservation resulted in a high color variance that on the training set, making this a very faulty algorithm. Therefore, a more efficient technique, based on tonal variation analysis, presented in [3] and [4] was used.

The object detection algorithm used required an automatic image segmentation associated with the *split & merge* method [3], applied on a quadtree [4]. From the detected objects, only those that verify a set of rules indicating a character are selected for the following stage.

The character recognition stage consists on the analysis of the geometric characteristics of each object. These characteristics (quadrant areas, perimeter, asymmetry, etc.) are used as sample data in the identification of alphanumeric characters. The process of identification was carried through using artificial intelligence techniques (automatic classifiers).

The second section in this paper will describe the first stage of the licenses plates' recognition process, the tracking of the license plate zone. The third section will be dedicated to the character detection and recognition algorithm. Finally, the fourth section will present the results obtained with the system prototype.

2. Tracking

The tracking of a car license plate is an important part in the architecture of the vehicle recognition system. The algorithm must give the plate's vertical and horizontal limits, so that the characters may be captured and recognized by an Optical Character Recognition (OCR) system.

The proposed algorithm is based on the analysis of the tonal variation on horizontal image lines. The aim of this analysis is to identify features that assure the exact tracking of the license plate. In order to increase the algorithm precision, the search area is divided into four zones. In each of these segments (beginning with the most probable zone) a search for consecutive horizontal lines which have a tonal pattern typical of a license plate is performed. This detection algorithm is divided into 4 stages which will be described next.

2.1 Detection of high contrast areas along image lines

Analyzing the color intensity along each line of the segment, figures 2.1 and 2.2, it is possible to notice that the license plate has a signature easily conclusive, due to the contrast between the license plate background and characters.

Fig.2.1 – Original image.

First is computed the color intensity 1st derivative along the line, in order to detect high contrast areas. The result for line A and B is showed in figure 2.2.a and 2.2.b. Objects with high contrast (peaks) will exhibit high or significant values on its edges, whereas objects with low contrast will tend to have zero. The color intensity variations with negative values take null values by convention, that is, the variations of light tones to dark tones were intentionally disregarded.

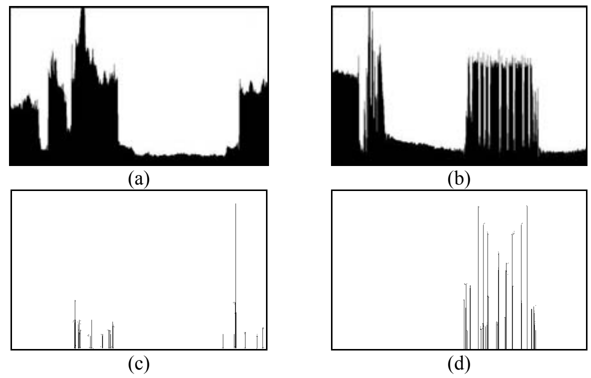


Fig.2.2 – Color intensity along: (a) line A, (b) line B;
1st derivative for color intensity along: (c) line A, (d) line B
(suppressed the less significant variations).

2.2 Line validation

After this pre-processing step, it's necessary to validate an image line as belonging to a license plate, in order to identify the plate's signature. The noise found in the images requires a selection of the peaks that are most probable to be part of a license plate signature.

This process consists on marking only the peaks that verify a predefined set of conditions directly related with the dimensions and color characteristics of the license plate. Among the considered conditions are the difference between the color intensity at the beginning and at the end of the variation and the width of each peak.

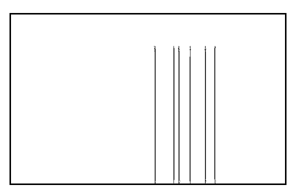


Fig.2.3 – Marking the variations that verify the conditions.

The group of defined peaks must then follow another set of rules that validates the line of the probable plate location. The values that must also be into expected intervals are: the number of marks; the distance between those marks. The result from this selection process applied to line A can be observed in figure 2.3.

2.3 License plate limits detection

The detection of the limits of license plate begins with the processing of the lines in each one of the four most probable zones of the captured image (center, lower left, lower right, top). The search for valid lines starts in the upper part of each zone to the bottom.

The tracking of the valid lines will consist in submitting each line to the validation process from up downwards of each zone. If the width of a group of validated lines corresponds to the expected average height of the characters, this group of lines will have a final approval. After this last appreciation, the license plate search ends.

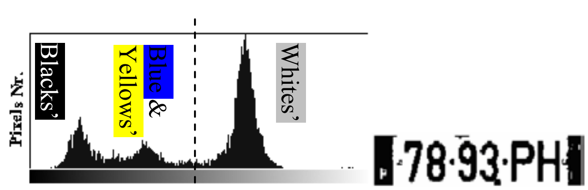
If the algorithm doesn't localize a group of lines recognizable as a license plate, it changes some of the parameters (number of lines, difference of intensity) in accordance with the results of the first search on image and begins another search, in order to refine and find the license plate. The number of attempts is pre-determined.

3. Characters Recognition

The following stage is the identification of the characters in the image. This process is divided into three steps: the detection of objects similar to characters; the computation of each object's characteristics; the objects identification, done by an automatic classifier trained for recognizing characters.

3.1 Objects detection

Once the tracking operation defines the most likely place for the license plate, an image color segmentation is executed in order to simplify the objects detection algorithm. Since the license plate is almost a black and white image, the color segmentation value is determined automatically through the C-Means algorithm [5][10][11]. This algorithm determines the color intensity level that defines the best split of the histogram into two uniform groups (black and white). The histogram and the segmented image can be seen in figures 3.1.a and 3.1.b.



(a) (b)

Fig.3.1 (a) License plate histogram after tracking; (b) segmented image.

After the image segmentation, the "split and merge" object detection algorithm - which uses the image quadtree representation - is applied.

The quadtree representation is created by dividing the image into four quadrants repeatedly, until they are only constituted by pixels with the same color, i.e., if the quadrant pixels aren't all from the same color, this zone is divided into another four smaller quadrants until the quadrants are unicolored (black or white). These unicolored quadrants are called *Leaves.*

For object detection, the "split and merge" algorithm is applied to the quadtree. This algorithm's basis is a search for neighbors that have the same color in the quadtree leaves, using an 8^{th} connectivity [6] (neighbors in N, NE, E, SE, S, SW, W, NW). The results are lists of neighbors that correspond to objects.

3.2 Objects characterization

Afterwards, an object characterization process is applied, triggering the object identification.

There are various features for each object that can distinguish the different alphanumeric characters. The computed characteristics are:

- Object areas in 4, 6 and 8 quadrants;
- Object's total area;
- Object's perimeter;
- Number of holes in the object;
- Holes' areas;
- Horizontal asymmetries (upper and lower part- Fig 3.3)
- Form factor.

Several tests are done to the object's size and location to eliminate the objects that are not characters of the license plate. For an object to be considered as a character it must have the following conditions: it must be black; its' height may not vary more than 5% of the next letter heights'; the subsequent character must always be at the right side of the previous one, without overlapping it in more than 5%.

All these object features must be normalized, in order to obtain common intervals, independently from the image size. Therefore, the features that have pixel*pixel units and the horizontal asymmetries are normalized by the image total area and the object perimeter by the image perimeter.

The quadrant and total areas before normalization are calculated by the sum of black pixels in each specified quadrant. The object perimeter is determined using the chain-code algorithm with an 8^{th} connectivity. The form factor is the relation between the maximum and minimum Feret diameters [7].

The horizontal asymmetry was chosen because it provides a mean to distinguish objects with similar quadrant areas. The horizontal asymmetry is calculated for the upper and lower parts of the image, from external or internal measurements. It is computed measuring each distance either from the center outwards, to the first black pixel found (for the internal asymmetry), or from the edge inwards, to the first black pixel found (for the external asymmetry). The final value is the sum of the differences between the individual distances calculated for the left and the right quadrant (see figure 3.2: A and B - external asymmetry; C and D - internal asymmetry) along the y axis in the upper part or the lower part.

The following expression represents the horizontal asymmetry for the upper part.

$$A_{ext} = \sum_{i=0}^{heigth/2} |a_i - b_i| ; \qquad A_{\text{int}} = \sum_{i=0}^{height/2} |c_i - d_i| ;$$

with: a_i, b_i – individual external distances for each line in the left and right quadrant, respectively;
c_i, d_i – individual internal distances for each line in the left and right quadrant, respectively;

The equivalent expression for the lower part is similar, except for the sum limits (in this case, from $heigth/2$ to *height*).

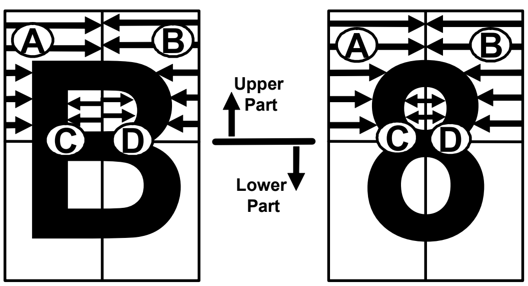


Fig 3.2 - Horizontal asymmetry differences for two similar characters

3.3 Object Identification

The information about each of the detected objects is presented to an intelligent classifier for identification. For the classifier design, several techniques, like nearest neighbor, neural networks or decision trees could be used [12][14].

In this project, the decision went for the decision trees architecture, using CART software, because of its readability and simplicity. One disadvantage, although, is that this technique always produces a result within the normal alphabet even if the object isn't a character. The possibility of having an undefined class would help, and it's a case of study for further development.

The Portuguese plates can have one of two configurations: four numbers in the left side followed by two letters or, for the older ones, two letters in left side followed by four numbers.

A pre-classification was executed, to reduce the process complexity and the number of misclassifications,. The training set was first divided in two classes: numbers and letters, and one decision tree was trained for this class separation, i.e. the 'number or letter' decision tree is queried. If the character is a number, the features will enter the 'numbers' decision tree to recognize it, otherwise it's the 'letters' decision tree that classifies it.

The introduction of this pre-classification provided important information about the license plate organization, allowing *a priori* the choice of the character type under analysis. For the second classification step two other decision trees were trained for classifying exclusively letters or numbers.

The training set for both trees received all the object information described on the previous section. This set was composed by approximately 30 examples in good conditions, for each character. The result was a first classification tree with 70 terminal nodes, a 'numbers' decision tree with 22 terminal nodes and a 'letters' decision tree with 40 terminal nodes.

4. Application Example

The testing scenario of this methodology was a prototype that works standalone, of an automatic access control system for a real parking. The system should be composed by a position sensor, for detecting the car presence, an electric barrier gate, a digital camera and a personal computer. At the moment the project is just capturing images for offline processing.

The software SIREVIA, developed in Borland C++ Builder, is responsible for the capture of the image to detect the car license plate number, verify the car authorization and finally to trigger the opening of the gate if the person/car is certified.

This program uses a local database, implemented in Microsoft Access, which gathers information about users and cars. The authorization is determined by the user category and for some categories by the current hour and day of week. Here is also stored a log system used mostly for statistics purposes.

4.1 Results

A test was done using photos caught in front of the vehicle, with one normal digital camera with the image sizes configured for 640x480 pixels. The captures were done with a good luminosity and with a maximum rotation of 30º degrees. The test set was composed by 250 images of normal car plates. The results for the accepted photos were:

96.7% of success on license plates tracking;

97.8% of success on correct color segmentation;

93.7% of success on individual characters recognition.

For the correctly tracked and segmented license plates, the results were: 86,3% of success on the global vehicles' license plates recognition.

The average recognition time for each photo was 109 hundreds of seconds (in an Intel Pentium IV at 1700 MHz).

4.2 Comparison with Similar Systems

In order to obtain a correct evaluation of the SIREVIA performance, several similar systems were analysed and results compared.

An older system, developed by UTAD [7], resulted in 100% correct trackings for stopped vehicles, 99,5% and 88,5% in moving vehicles, with good and bad climatic conditions, respectively. The OCR of UTAD project obtained 98,4% of success on segmented characters with the use of critical points and template matching methods. Since computation times were not published, we can only compare the accuracy that is similar in both systems.

The SIAV system [9] obtained 98,7% of success in plates' localization and 87,1% in correct segmentation of each character, using a test set of 300 images. Furthermore, it achieved 82,4% in tracking with the tonal variation algorithm and 87% in the recognition of individual segmented characters. The total correct recognition of license plates' (7 digits) had been of 28%, and the computation time was between 1,4 and 130 seconds (in a AMD K6-2 400 MHz). Compared with this system, SIREVIA achieved faster computation time and better accuracy.

5. Conclusions

A vehicle recognition system based on license plate identification was proposed using image processing techniques and artificial intelligent techniques. The methodology is divided in two stages: the tracking and the character detection and recognition. The total execution time is about 1s for each image, with a 96.7% of success in the tracking stage and 86.3% of success on global license plate recognition, which can be considered quite good results, considering the image complexity.

The comparison with similar systems also has shown that SIREVIA presents a good compromise between speed and accuracy.

Future work will be dedicated to improve the accuracy and computation times, mainly on the character detection and recognition stage. An improved character detection algorithm using color information and new object features are planned as next improvements for the near future.

References:

[1] "*London Introduces Controversial City Driving Tax*", Voa News, London, Feb.17/02/ 2003

[2] Transports for London, "*Annual Report by the Impacts Monitoring Group in Congestion Charging Division of Transport for London*", ISBN 1-871829-18-6.

[3] Ueda, N., Nakano, R., Ghahramani, Z., and Hinton, G. E.,"*Split and merge EM algorithm for improving Gaussian mixture density estimates (invited)*", Journal of VLSI Signal Processing Systems, Vol. 26, pp. 133-140, 2000.

[4] H.Samet, "*The quadtree and related hierarchical data structures*", ACM Computing Surveys 16: 188–260, 1984.

[5] Y.W. Lim, S.U. Lee, "*On the color image segmentation algorithm based on the thresholding and the fuzzy c-means techniques*". Pattern Recognition, 23(9):935–952, 1990.

[6] Gonzalez R., Woods R., "*Digital Image Processing*", Addison-Wesley Publishing Company, 1992.

[7] Barroso J., Rafael A., et al, "*Number plate reading using computer vision*", Industrial Electronic ISIE'97, Proceedings of IEEE International Symposium on Publish, 1997.

[8] See/Car Technical Information, *Hi-Tech Solutions Ltd.* - Gabriel Industrial Park POBox #133 Migdal-Haemek Israel - www.htsol.com

[9] Coelho de Souza, et al, "*SIAV - Um Sistema de Identificação Automática Veículos*", XIII Cong. Brasileiro de Automática - CBA2000, p.1377-1380, 2000.

[10] Torsten Seemann, "*Digital Image Processing Using Local Segmentation*", PhD Thesis, School of Computer Science and Software Engineering. Australia, FIT - Monash University, 2002.

[11] M. Takatoo *et al.*, "*Gray scale image processing technology applied to vehicle license number recognition system*," in Proc. Int. Workshop Industrial Applications of Machine Vision and Machine Intelligence, 1987, pp. 76–79.

[12] V. Wu, R. Manmatha, E. Riseman, "*TextFinder:An automatic system to detect and recognize text in images*", IEEE Trans. Pattern Anal. Machine Intell., vol. 21, no. 11, Nov. 1999.

[13] B. Jaehne, *Practical Handbook on Image Processing for Scientific Applications*. Boca Raton, FL: CRC Press, 1997.

[14] Rainer Lienhart, Axel Wernicke, "*Localizing and Segmenting Text in Images and Videos*", IEEE Transactions On Circuits And Systems For Video Technology, vol. 12, no.4, April 2002.