Automatic Number Plate Recognition System

Hakob Sarukhanyan, Souren Alaverdyan, and Grigor Petrosyan
Institute for Informatics and Automation Problems of NAS RA, Yerevan, Armenia
hakop,souren,grigor@ipia.sci.am

ABSTRACT
The software system of plate identification and recognition from video images is presented in this paper.

Keywords
License plate, Hough transform, digital filter, optical character recognition, chain code.

1. INTRODUCTION
The automatic detection and recognition of car number plates has become an important application of artificial vision systems [1-8]. The object is to develop a system whereby cars passing a certain point are digitally photographed, and then identified electronically by locating the number plate in the image, segmenting the characters from the located plate and then recognizing them. Some applications for a number plate recognition system are: 1) Traffic flow measurement and planning; 2) Tracking stolen vehicles; 3) Control and security at tolling areas, e.g. parking garages; 4) Traffic law enforcement (automatically identifying speed ers, illegal parking, etc.); 5) The system could also be adapted for use in reading e.g.; 6) Warehouse box stencil codes; 7) Train rolling stock codes; 8) Aircraft tailcodes.

This system consists of two high-level stages: In the first stage, the number plate is detected and segmented from a digital image of the car being examined. The plate location and size are passed to the optical character recognition (OCR) subsystem.

Fig. 1 shows the proposed license plate recognition process. There are six primary algorithms that the software requires to identify a licence plate:
1. Plate localisation – responsible for finding and isolating the plate on the picture;
2. Plate orientation and sizing – compensates for the skew of the plate and adjusts the dimensions to the required size;
3. Normalisation – adjusts the brightness and contrast of the image;
4. Character segmentation -finds the individual characters on the plates;
5. Optical character recognition;
6. Syntactical/Geometrical analysis – check characters and positions against country specific rules

The methods are all based on several assumptions concerning the shape and appearance of the license plate. The assumptions are listed below:

a) The license plate is a rectangular region of an easily discernable color;
b) The width-height relationship of the license plate is known in advance;
c) The orientation of the license plate is approximately aligned with the axes;
d) Orthogonality is assumed, meaning that a straight line is also straight in the image and not optically distorted.

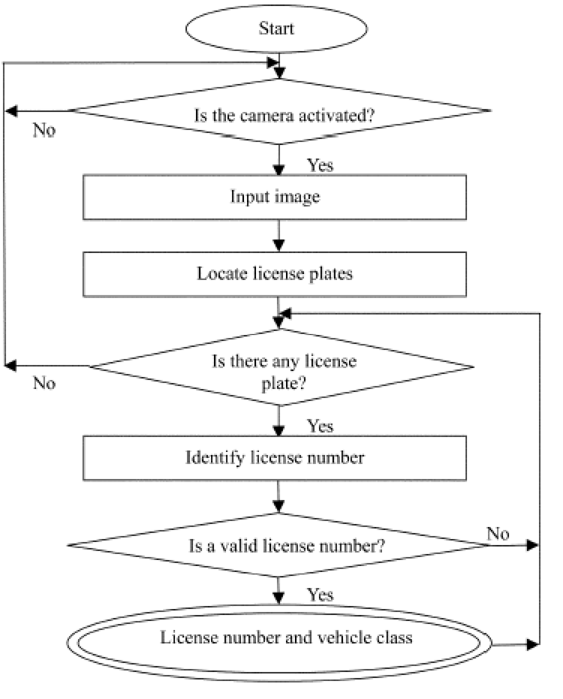


Fig 1: Diagram of the proposed LPR process.

2. EXTRACTING LICENSE PLATES BY HOUGH TRANSFORM

This section presents a method for extracting license plates based on the Hough transform. The first step is to threshold the gray scale source image. Then the resulting image is passed through two parallel sequences, in order to extract horizontal and vertical line segments respectively. The first step in both of these sequences is to extract edges. The result is a binary image with edges highlighted. This image is then used as input to the Hough transform, which produces a list of lines in the form of accumulator cells. These cells are then analyzed and line segments are computed. Finally the list of horizontal and vertical line segments are combined and any rectangular regions matching the dimensions of a license plate are kept as candidate regions. This is also the output of the algorithm.

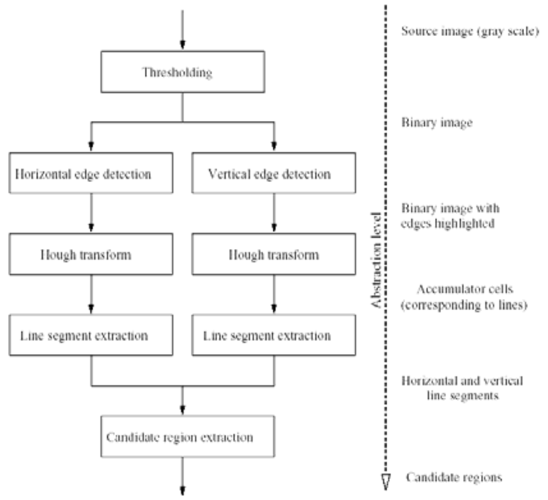


Fig. 2: Overview of the Hough method

As shown in Figure 2 the algorithm behind the method consists of five steps. The first step is to detect edges in the source image. This operation in effect reduces the amount of information contained in the source image by removing everything but edges in either horizontal or vertical direction. This is highly desirable since it also reduces the number of points the Hough transform has to consider. The edges are detected using spatial filtering. The choice of kernels was partly based on experiments and partly because they produce edges with the thickness of a single pixel, which is desirable input to the Hough transform.

3. LINE DETECTION USING HOUGH TRANSFORM

The Hough transform is a method for detecting lines in binary images. The method was developed as an alternative to the brute force approach of finding lines, which was computationally expensive ($O(n^3)$). In contrast the Hough transform performs in linear time. The Hough transform works by rewriting the general equation for a line through (x_i, y_i) as:

$$y_i = ax_i + b \Rightarrow b = -ax_i + y_i \qquad (1)$$

For a fixed (x_i, y_i), Eq. (1) yields a line in parameter space and a single point on this line corresponds to a line through (x_i, y_i) in the original image. Finding lines in an image now simply corresponds to finding intersections between lines in parameter space. In practice, instead the Eq. (1) is used the following:

$$x\cos\mu + y\sin\mu = r \qquad (2)$$

In Eq. (2) the parameter μ is the angle between the normal to the line and the x-axis and the r parameter is the perpendicular distance between the line and the origin. This is also illustrated in Fig. 3. Also in contrast to the previous method, where points in the image corresponded to lines in parameter space, in the form shown in Eq. (2) points correspond to sinusoidal curves in the (r, μ) plane.

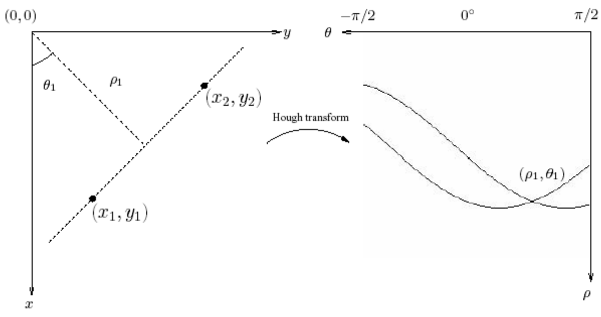


Fig. 3: Example image and corresponding Hough transform

Fig. 3 shows two points, (x_1, y_1) and (x_2, y_2), and their corresponding curves in parameter space. As expected, the parameters of their point of intersection in parameter space correspond to the parameters of the dashed line between the two points in the original image. In accordance with the preceding paragraph, the goal of the Hough transform is to identify points in parameter space, where a high number of curves intersect. Together these curves then correspond to an equal amount of collinear points in the original image. A simple way to solve this problem is to quantize the parameter space. The resulting rectangular regions are called accumulator cells and each cell corresponds to a single line in the image. The algorithm behind the Hough transform is now straight forward to derive. First the accumulator array is cleared to zero. Then for each point in the edge image iterate over all possible values of μ and compute $r = 1/2$ using Equation (2). Finally for each computed $(1/2, \mu)$ value, increment the corresponding accumulator cell by one. Since the algorithm iterates over all points it is clear that it performs in O(n) time.

4. PLATE CANDIDATES RECTANGLES FILTERING

After identification of plate candidates we must realize some special filters before optical character recognition stage. Below we give a short description of filtering with real plate candidate example (see Fig. 4).

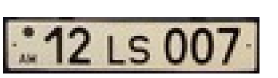


Fig.4. Plate candidate

Step 1. Apply low frequency filter with convolution matrix

$$H = \frac{1}{10}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

The result of this filter given in Fig. 5.

Fig.5. Plate after convolution

Step 2. Apply the hyperbolic sinusoidal filter for brightness enhancement: $sh(x) = (e^{x} + e^{-x})/2$, where $x \in [0,1]$ is a normalized pixel value of plate image. The result given in Fig.6.

Fig. 6. The result of $sh(x)$ filter.

Step 3. Apply the homogeneity enhancement filter $h_1(x) = (x/k)^{x}$, where $x \in [0,1]$, $k = 3,4,\ldots,20$.

Fig. 7. The result of $h_1(x)$ filter.

Step 4. Apply image binarization filter

$$x^{*}_{i,j} = \begin{cases} 1, & if \quad x_{i,j} \le s_{i,j}, \\ 0, & otherwise, \end{cases}$$

where the threshold $s_{i,j}$ is defined as follows

$$s_{i,j} = \frac{1}{25} \sum_{p=i-2}^{i+2} \sum_{q=j-2}^{j+2} x_{p,q}, \quad x_{p,q} \in [0,255]$$

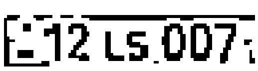


Fig. 8. The result of $h_1(x)$ filter.

Step 5. Apply the thinning filter over the binary image to obtain the contour with one pixel thickness [pratt]. The result of this filter given in Fig. 9.

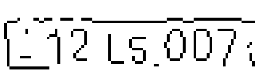


Fig. 9. The result of thinning filter.

Step 6. Using the vertical and horizontal projection removes the false piece on the plate and defines also the number of rows and symbols in the plate.

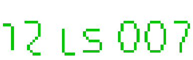


Fig. 10. The result after removing false pieces.

Step 7. Defining the chain code of symbols and its recognition. To create chain code of symbols from Fig. 10 at first symbols are approximated by linear segments

Fig. 11. Approximation by linear segments

as shown in a Fig. 11, then using basic directions (see Fig. 12) we obtain appropriate chain codes of symbols.

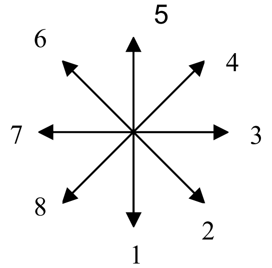
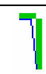
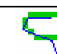


Fig. 12. Basic directions schemes

The appropriate chain codes for considering example are given in the table below

Symbol number	Symbol	Chain code
1		31
2		31823
3		13
4		82173
5		1357
6		1357
7		38

Step 8. Symbol recognition. Thus, each symbol in the plate is associated with chain code. Note that these codes can be different for the same symbol. Moreover, the same chain code can be associated to different symbols. This phenomenon is called *collision*. Note that to provide symbol recognition uniqueness some additional parameters, such as segment slope angle, symbol height, the index in the row, and etc., are calculating when construct chain code.

For example, the first symbol “1” got chain cod 31, the last one “7” got chain code 38. If we don’t consider segment vertical slope value of symbol “7”, we have chain code 31, so collision appear (symbols “1” and “7” got identical chain code 31). If considering segment slope parameter, symbol “7” gets chain code 38, but not 31. It should be noted that everything depends on scanned image quality. If symbols’ parameters are very fewely different each of other, then the collision can appear.

5. ACKNOWLEDGEMENT

The authors would like to thank ISTC for supporting this research around the project A-1451.

REFERENCES

[1] R. Gonzalez, R. Woods R., *Digital Image Processing*, Prentice Hall, New Jersey, 2002

[2] V. Shapiro, D. Dimov, S. Bonchev, V. Velichkov, G. Gluhchev, "*Adaptive License Plate Image Extraction"*, International Conference Computer Systems and Technologies, Rousse, Bulgaria, 2004

[3] Y. Zhang, C. Zhang, *"New Algorithm for Character Segmentation of License Plate"*, Intelligent Vehicles Symposium, IEEE, 2003

[4] Shyang-Lih Chang, Li-Shien Chen, Yun-Chung Chung, Sei-Wan Chen, "*Automatic License Plate Recognition*", IEEE Trans. On Intelligent Transformation Systems, vol. 5, No. 1,2004

[5] Y. Cui, Q. Huang, "*Character extraction of license plates from video*", IEEE Conf. Computer Vision and Pattern Recognition, 1997, pp. 502–507.

[6] D. S. Gao, J. Zhou, "*Car license plates detection from complex scene*," Proc. 5th Int. Conf. Signal Processing, vol. 2, 2000, pp. 1409–1414.

[7] S. K. Kim, D. W. Kim, H. J. Kim, "*A recognition of vehicle license plate using a genetic algorithm based segmentation*," Proc. Int. Conf. Image Processing, vol. 2, 1996, pp. 661–664.

[8] Charl Coetzee, Charl Botha, David Weber. "*PC Based Number Plate Recognition System*", IEEE, 1998