

# Building an Automatic Vehicle License-Plate Recognition System

Tran Duc Duan<sup>(1)</sup>, Tran Le Hong Du<sup>(1)</sup>, Tran Vinh Phuoc<sup>(2)</sup>, Nguyen Viet Hoang<sup>(1)(3)</sup>

{tdduan, tlhdu}@fit.hcmuns.edu.vn, phuoc.gis@ditagis.org, hnguyen5@uiuc.edu

**Abstract**—Due to a huge number of vehicles, modern cities need to establish effectively automatic systems for traffic management and scheduling. One of the most useful systems is the Vehicle License-Plate (VLP) Recognition System which captures images of vehicles and read these plates’ registration numbers automatically. In this paper, we present an automatic VLP Recognition System, *ISeeCarRecognizer*, to read Vietnamese VLPs’ registration numbers at traffic tolls. Our system consists of three main modules: VLP detection, plate number segmentation, and plate number recognition. In VLP detection module, we propose an efficient boundary line-based method combining the Hough transform and Contour algorithm. This method optimizes speed and accuracy in processing images taken from various positions. Then, we use horizontal and vertical projection to separate plate numbers in VLP segmentation module. Finally, each plate number will be recognized by OCR module implemented by Hidden Markov Model. The system was evaluated in two empirical image sets and has proved its effectiveness (see section IV) which is applicable in real traffic toll systems. The system can also be applied to some other types of VLPs with minor changes.

**Index Terms**—Vehicle License-Plate Recognition, Real-time System, Hough Transform, Contour Algorithm.

## I. INTRODUCTION

THE problem of VLP recognition is a very interesting but difficult one. It is very useful for many traffic management systems. VLP recognition requires some complex tasks, such as VLP detection, segmentation and recognition. These tasks become more sophisticated when dealing with plate images taken in various inclined angles or plate images with noise. Because this problem is usually used in real-time systems, it requires not only accuracy but also fast processing. Most VLP recognition applications reduce the complexity by establishing some constrains on the position and distance from the camera to vehicles, and the inclined angles. By that way, the recognition rate of VLP recognition systems has been improved significantly. In addition, we can gain more accuracy by using some specific features of local VLPs, such as the number of characters, the number of rows in a plate, or colors of plate background, or the ratio of width to height of a plate (see Fig 1.).



Fig 1. Types of Vietnamese plates: 1 row and 2 rows

## II. RELATED WORK

The problem of automatic VLP recognition has been studied since 1990s. The first approach was based on characteristics of boundary lines. The input image was first processed to enrich boundary lines’ information by some algorithms such as the gradient filter, and resulted in an edging image. This image was binarized and then processed by certain algorithms, such as Hough transform, to detect lines. Eventually, couples of 2-parallel lines were considered as a plate-candidate [4][5]. Another approach was morphology-based [2]. This approach focuses on some properties of plate images such as their brightness, symmetry, angles, etc. Due to these properties, this method can detect the similar properties in a certain image and locate the position of license plate regions. The third approach was texture-based [3]. In this approach, a VLP was considered as an object with different textures and frames. The texture window frames of different sizes were used to detect plate-candidates. Each candidate was passed to a classifier to confirm whether it is a plate or not. This approach was commonly used in finding text in images tasks. In addition, there have been a number of other methods relating to this problem focusing on detecting VLP in video data (objects appear in a chain of sequent images)[6][7].

## III. THE PROPOSED SYSTEM

Our system, *ISeeCarRecognizer*, consists of four modules: Pre-processing, VLP detection, character segmentation, and optical character recognition (OCR), in which the last three modules deal with three main problems of a VLP recognition domain.

(1) Faculty of Information Technology, University of National Sciences, Vietnam National University of Hochiminh city, Hochiminh, Vietnam.

(2) Center for Information Technology and GIS - DITAGIS, HCM City University of Technology, Vietnam.

(3) University of Illinois at Urbana Champaign, Illinois, USA.

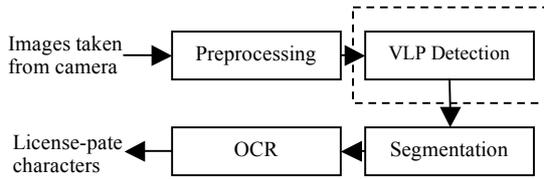


Fig 2. Main modules in ISeeCarRecognizer system

The *VLP detection module* receives images which have been processed by the preprocessing module – the first input module of this system. The resulted images of this module are sent to the segmentation module. The *segmentation module* segments plate-images into separate character-images. These character-images are then recognized by the *OCR module* and the final results are ASCII characters and numbers in plates (see Fig 2).

### A. Preprocessing

Images taken from camera were processed by the preprocessing module. The purpose of this module was to enrich the edge features. Because our detection method bases on the boundary features, it will improve the successful rate of the VLP detection module. The algorithms sequentially used in this module are graying, normalizing and histogram equalization. After having obtained a grey-scale image, we use Sobel filters to extract the edging image, and then thresholds the image to a binary one. We used the local adaptive thresholding algorithm for the binarization step. Especially, we develop an algorithm based on dynamic programming to optimize its speed and make it suitable to real-time applications [1]. The resulted images are used as inputs for the VLP detection module.

### B. VLP Detection Algorithm

In boundary-based approach, the most important step is to detect boundary lines. One of most efficient algorithms is Hough transform applying to the binary image to extract lines from object-images. Then we look for two parallel lines, whose the contained region is considered plate-candidates. However, the drawback of this approach is that the execution time of the Hough transform requires too much computation when being applied to a binary image with great number of pixels. Especially, the larger image the slower the algorithm is. The speed of the algorithm may be improved by thinning image before applying the Hough transform. Nevertheless, the thinning algorithm is also slow. This limitation makes the approach unsuitable for real time traffic management systems.

The algorithm we used in this system is the combination of the Hough Transform and Contour algorithm which produces higher accuracy and faster speed so that it can be applied to real time systems.

#### 1) Combine Hough Transform and Contour Algorithm for Detecting VLP

Our approach is as follows: from the extracted edging image, we use the contour algorithm to detect closed boundaries of objects. These contour lines are transformed to Hough coordinate to find two interacted parallel lines (one of 2-parallel lines holds back the other 2-parallel lines and establishes an parallelogram-form object) that are

considered as a plate-candidate. Since there are quite few (black) pixels in the contour lines, the transformation of these points to Hough coordinate required much less computation. Hence, the speed of the algorithm is improved significantly without the loss of accuracy (see Fig 3).

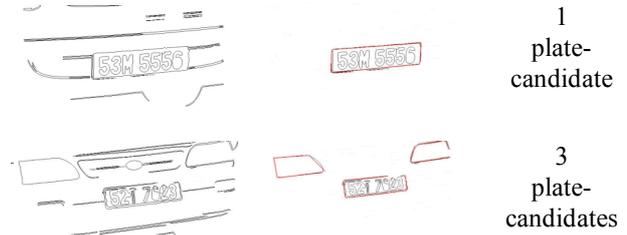


Fig 3. Two successful cases when use combination of Contour algorithm and Hough transformation

However, some plates may be covered by glasses or decorated with headlights. These objects may also have the shape of two interacted 2-parallel lines, and therefore, are also falsely detected as plate-candidates. To reject such incorrect candidates, we implement a module for evaluating whether a candidate is a plate or not.

#### 2) Plate-Candidates Verification

From the two horizontal lines of a candidate, we can calculate exactly how inclined it was from horizontal coordinate. Then we apply a rotate transformation to adjust it to straight angle. After processed, these straight binary plate-candidate regions were passed to a number of heuristics and algorithms for evaluating.

Our evaluating plate-candidates algorithm bases on two main steps, which are taken respectively. The two steps are: (a) evaluate the ratios between the heights and the widths of the candidates, (b) use horizontal crosscuts to count the number of cut-objects in the candidates.

##### a. Evaluate the ratios between the widths and the heights of the candidates

In this stage, we check and only select out candidates that have the ratios of width to height satisfying pre-defined constraint:

$$\min\text{WHRatio} < W/H < \max\text{WHRatio}$$

Since there are two main types of Vietnamese plates: 1-row and 2-row (See Fig 2), we have two adequate constraints for two types.

$$3.5 < W/H < 4.5 \text{ with 1-row plate-candidates}$$

$$0.8 < W/H < 1.4 \text{ with 2-row plate-candidates}$$

Those candidates which satisfied one of the two above constraints are selected and passed to the next evaluation.

##### b. Evaluate by using horizontal crosscuts

In this stage, we use two horizontal cuts and then count the number of objects that are cut by these crosscuts. A candidate will be considered as a plate if the number of cut objects is in the given range chosen suitably for each plate type by experiments (see examples in Table 1). This number must be in the approximate range of the number of characters in a VLP, we have two appropriate constraints for two types of Vietnamese plates:

$4 \leq N \leq 8$  with 1-row plate-candidates

$7 \leq N \leq 16$  with 2-row plate-candidates

With  $N$  is the number of cut-objects.

The candidates that satisfied one of the two above constraints are selected as the final result.

Plate-candidates	Number of objects	Result
	2	Non-plate
	2	Non-plate
<del>53M-4893</del>	10	Plate
<del>52T</del> <del>7013</del>	4	Plate

Table 1. A sample of using horizontal cuts to evaluate plate-candidates

In our system, we implemented two horizontal cuts at 1/3 and 2/3 of plate-candidate's height. The average of number of cut objects will be calculated. This evaluation helps to identify the correct plate-candidates.

### C. Segmentation

To correctly recognize characters, we have to segment a binary plate image to set of images which only contain one license character. These character images will be passed to the OCR module for recognizing. The common algorithm for this task is applying projections. However, in some cases, it does not work correctly. We will now describe our approach in segmentation by adding some enhancements to this method.

We use a horizontal projection to detect and segment rows in 2 row plates. Because binary plate images were adjusted their inclined angles to zero, the result of row segmentation is nearly perfect. The positions with minimum values of horizontal projection are the start or the end of a row in plate.

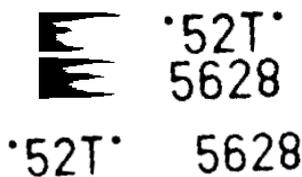
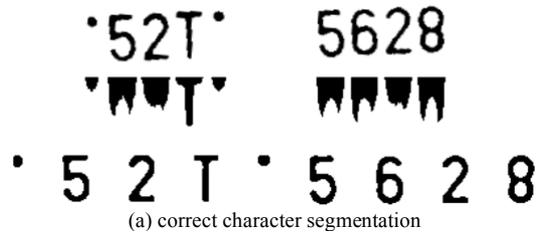
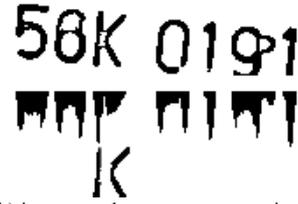


Fig 4. Results of row segmentation by horizontal projection.

Different from row segmentation, character segmentation is more difficult due to many reasons such as stuck characters, screws, and mud covered in plates. These noise things cause the character segmentation algorithm using vertical projection to have some mistakes. In some worst cases of bad quality plate images, a character can be segmented into two pieces. We apply several constraints of ratio of the height to the width of a character.



(a) correct character segmentation



(b) incorrect character segmentation

Fig 5. Character segmentation by vertical projection.

We search for the minimum values in the vertical projection and only the minimum positions which give cut pieces satisfied all predefined constraints are considered as the points for character segmentation. By this enhancement, we have achieved better results in this task. After this step, we have a list of character candidates. Not all of the candidates are actually images of characters. For example in Fig 5(a), we have 9 candidates but only 7 of them are character images. The first and third candidates are images of 2 screws in the plate.

By that time, we can re-evaluate whether a plate candidate is a plate or not by checking the number of characters of candidates. In Vietnam, a plate contains only 7 or 8 characters (see Fig 1). The final plate candidates, together with their list of characters are passed to the OCR module for recognizing.

### D. Hidden Markov Model for OCR

In this model, all images used for training and images to deal with after well trained are the same size of 50x50 pixels. Hence, all character images are scaled into the size of 50x50 pixels by the centre point.

#### 1) Features extracting

In this system, we use the HMM model for character recognition. The features which we used in this model are the ratio of foreground pixels in a window. Each window gives us a value by following formula:

$$f_k = \frac{\sum_{(i,j) \in S_k} foreground\_pixels(i,j)}{Width_k \times Height_k}$$

We use a window with the size of 9x9, and scan this window in the image from left to right and top to bottom (see Fig 6). These windows can overlap each other by two thirds of their size. By this way, we have a feature vector which includes 196 values.

$$F = \{f_1, f_2, \dots, f_{196}\}$$

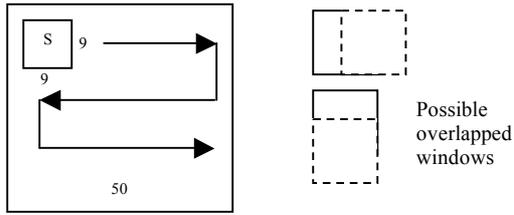


Fig 6. Scanning a window in an image and overlapped windows

## 2) Training model

In the recognition module, we need to classify a character image into one of 36 classes (26 alphabet letters: A, B, C... and 10 numeric characters: 0, 1, 2...). To train our model, we use training sets which were extracted from images of VLPs. The number of samples for every class is about 60. These samples were extracted from real VLP images with a little noise, so after well trained, the model can recognize exactly plates with the similar types of noise. In the last step, we use some specific rules of Vietnamese VLPs to improve accuracy. We learned that the third character in plate must be a letter, the fourth is sometimes a letter but usually a number, and the other positions are surely numbers.

## IV. EMPIRICAL EVALUATION

Our system was evaluated with two sets of Vietnamese vehicles' plates. Images were taken by a Sony DC350 digital camera, with size of 800x600 pixels, in different places and times. We use Microsoft Visual C++ 6.0, run on HP Workstation X2000 Pentium IV, 1.4 GHz, 512 MB RAM, Windows XP OS.

Image set	Camera position	Light condition	Number of images
A	Airport check-in office. Rotated angles: right, left ~ 30°	10-12 A.M	415
B	Random locations. Rotated angles: right, left ~ 30° or straight	Morning or night (with flash light)	390

Table 2. Image sets were used in tests

Evaluating was taken on the two image sets above and showed the following results:

Image set	A	B	A+B
Number of images	415	390	805
Corrects	412	383	795
Errors	3	7	10
Correct rate	99.27%	98.2%	98.76%
Error rate	0.73%	1.8%	1.24%
Avg execution time per image	0.65s	0.65s	0.65s

Table 3. Test results of VLP detection module

Image set	A	B	A+B
Number of VLP images	412	383	795
Corrects	404	372	776

Errors	8	11	19
Correct rate	98.05%	97.13%	97.61%
Error rate	1.95%	2.87%	2.39%
Avg execution time per candidate	0.1s	0.1s	0.1s

Table 4. Test results of character segmentation module

Image set	A	B	A+B
Number of VLPs	412	383	795
Number of characters	2987	2780	5767
Corrects	2922	2702	5624
Errors	65	78	143
Correct rate	97.82%	97.19%	97.52%
Error rate	2.18%	2.81%	2.48%

Table 5. Test results of OCR module

Image set	Module	Correct	Error
A	VLP Detection	99.27%	0.73%
	Segmentation	98.05%	1.95%
	OCR	97.82%	2.18%
B	VLP Detection	98.2%	1.8%
	Segmentation	97.13%	2.87%
	OCR	97.19%	2.81
A+B	VLP Detection	98.76%	1.24%
	Segmentation	97.61%	2.39%
	OCR	97.52%	2.48%
	<b>Whole system</b>	<b>92.85%</b>	<b>7.15%</b>

Table 6. Test results of whole system

## V. CONCLUSIONS AND FUTURE WORK

### A. Conclusions

The system performs well on various types of Vietnamese VLP images, even on scratched, scaled plate images. In addition, it can deal with the cases of multiple plates in the same image, or different types of vehicles such as motorbike plates, car plates or truck plates. However, it still has a few errors when dealing with bad quality plates.

### B. Future work

We are working on a number of algorithms in the preprocessing module. The purpose is to detect regions that are likely plate regions first and thus to reduce the computation cost of the VLP detection algorithm. In addition, we intend to combine a number of texture-based approaches, and machine learning methods to evaluate plate-candidates. We believe these will improve the accuracy and the speed of the algorithm furthermore.

## REFERENCES

- [1] D. A. Duc, T. L. Du, T. D. Duan, "Optimizing Speed for Adaptive Local Thresholding by using Dynamic Programming, The 7<sup>th</sup> International Conference on Electronics, Information, and Communications, pp438-441, Vol 1, 2004.
- [2] Jun-Wei Hsieh, Shih-Hao Yu, Yung-Sheng Chen, "Morphology-based License Plate Detection from Complex Scenes", 16th International Conference on Pattern Recognition (ICPR'02) Vol 3, 2002.
- [3] Kwang In Kim, Keechul Jung and Jin Hyung Kim, "Color Texture-based Object Detection: an Application to License Plate Localization", Lecture Notes in Computer Science, pp. 293-309, Springer, 2002.
- [4] Kamat, Varsha, and Ganesan, "An Efficient Implementation of the Hough Transform for Detecting Vehicle License Plates Using DSP'S", Proceedings of Real-Time Technology and Applications, pp58-59, 1995.
- [5] Yasuharu Yanamura, Masahiro Goto, Daisuke Nishiyama, "Extraction and Tracking of the License Plate Using Hough

Transform and Voted Block Matching”, IEEE IV2003 Intelligent Vehicles Symposium Conference, 2003.

- [6] A. Rahman, Ahmad Radmanesh, “A Real Time Vehicle’s License Plate Recognition”, Proceedings of the IEEE on Advanced Video and Signal Based Surveillance, AVSS’03, 2003.
- [7] K. Kanayama, Y. Fujikawa, K.Fujimoto, and M.Horino, “Development of Vehicle-License Number Recognition System Using Real-Time Image Processing and Its Application to Tracal-time measurement”, In Proceedings of IEEE Vehicular Technology Conference, pp798-804, 1991.