Segmentation of Characters on Car License Plates

Xiangjian He[#1], Lihong Zheng[*2], Qiang Wu[#1], Wenjing Jia[#1], Bijan Samali[#3] and Marimuthu Palaniswami[&4]

[#] *Faculty of Engineering and Information Technology*
University of Technology, Sydney
PO Box 123, Broadway
NSW 2007, Australia
[1]{sean,wuq,wejia}@it.uts.edu.au
[3] bijian@eng.uts.edu.au

[*] *Charles Sturt University*
Wagga Wagga Campus
NSW, Australia
[2] lzheng@csu.edu.au

[&] *Department of Electrical and Electronic Engineering*
The University of Melbourne
Victoria, 3010, Australia
[4] swami@ee.unimelb.edu.au

Abstract—**License plate recognition usually contains three steps, namely license plate detection/localization, character segmentation and character recognition. When reading characters on a license plate one by one after license plate detection step, it is crucial to accurately segment the characters. The segmentation step may be affected by many factors such as license plate boundaries (frames). The recognition accuracy will be significantly reduced if the characters are not properly segmented. This paper presents an efficient algorithm for character segmentation on a license plate. The algorithm follows the step that detects the license plates using an AdaBoost algorithm. It is based on an efficient and accurate skew and slant correction of license plates, and works together with boundary (frame) removal of license plates. The algorithm is efficient and can be applied in real-time applications. The experiments are performed to show the accuracy of segmentation.**

I. Introduction

License Plate Recognition (LPR) has found numerous applications in various areas [1, 2].

LPR system usually consists of three major components: license plate detection, character segmentation and license plate (or character) recognition.

License plate detection is an important step in a LPR system. The quality of a license plate detector influences the accuracy of license plate recognition. On the other hand, many factors can affect the accuracy and efficiency of license place detection. Most existing license plate detection algorithms are restricted by various controlled conditions such as fixed backgrounds, known color, or designated ranges of the distance between cameras and vehicles. In this paper, we use an AdaBoost learning algorithm as shown in [3] based on both global statistical features and local Haar-features to construct a cascaded classifier for license plate detection.

There have been various commercial systems for license plate recognition around the world. Among these systems, two types of classifiers are applied. They are (optical character recognition) OCR-based method and learning-based method.

OCR has shown its advantage in recognizing printed document or text where the background has no or very little noise. However, license plate images captured in real-time usually contain heavy noise and are with complex backgrounds. OCR based recognition requires image pre-processing to remove the boundaries before it can be properly used for license plate recognition under complex and unrestricted conditions [4]. To the best knowledge of authors, there are no papers around which have shown clear, efficient and successful techniques for satisfactorily removing the boundaries of license plates or for accurate segmentation of license plates from their backgrounds.

Learning-based approach such as those shown in [5, 6] extracts optimal features of characters to improve the recognition accuracy. This approach heavily relies on the accurate character segmentation. However, license plate images taken in real-time can be very difficult for proper character segmentation due to image noise, plate frame, rivet, space mark, plate rotation and illumination variance [7]. Improperly segmented characters will result in misrecognized characters.

In recent years, there have been many algorithms developed for character segmentation on license plates. These algorithms include the works shown in [7] based on Hough transform, and in [8] based on horizontal and vertical projections. However, it is still a problem for accurate and real-time character segmentation under the situations when license plate boundaries are connected to inside characters, characters are connected to each other, and characters are broken [9].

In this paper, we present an approach that can correctly and efficiently segment the characters on license plates.

The rest of the paper is organized as follows. In Section 2, we give a brief overview of an algorithm for license plate detection and localization. The techniques for character segmentation are presented in Section 3. Experimental results are demonstrated in Section 4. The conclusions are made in Section 5.

II. LICENSE PLATE DETECTION

As shown in [3], the basic idea of the detection algorithm was to use a variable scanning window moving around on the input vehicle image. At each position, the image area covered by the scanning window was classified using a pre-trained classifier as either a license-plate area (a positive decision) or a non-license-plate area (a negative decision). The classifier used in this algorithm was a significant extension of Viola and Jones' work shown in [10] to license plate detection.

The algorithm adapted for license plate detection is as shown in [3], we construct a six-layer cascaded classifier to increase the detection speed, in which the first two layers are based on global features and the last four layers are based on local Haar-like features. A positive result from the first classifier triggers the evaluation of a second classifier. A positive result from the second classifier triggers a third classifier, and so on. A negative outcome at any layer leads to the immediate rejection of the image region (block). In other words, those image regions that are not rejected by the initial classifier will be processed by a sequence of classifiers. If any classifier rejects a selected image region, no further processing will be performed for this region. The reason of using this method for detection before recognition is that the algorithm can be achieved in real-time, accurately and can be applied for more flexible applications under complex environmental conditions.

When the detected images have been rotated by a certain angle (less than 45^o in both directions) and skewed by a degree, we perform a skew and slant correction using the method based on a modified Hough transform as shown in [11] when needed.

Fig. 1 shows two license plates detected using the algorithms introduced in this section (but without skew and slant correction as the rotation angles and skew degrees of these two images are small and do not affect our segmentation performance).

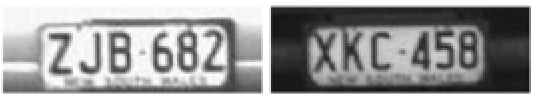


Fig. 1. License plate images detected

III. CHARACTER SEGMENTATION

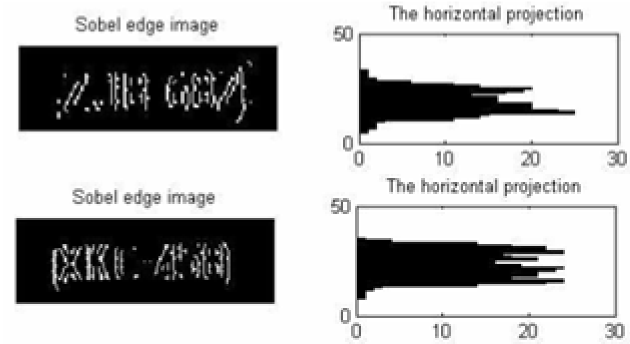
The steps for character segmentation include vertical edge detection together with horizontal projection for upper and lower boundary removal, and image binarization together with vertical projection to find the segmenting points.

A. Removing upper and lower bounds of license plates

This step contains two parts: vertical edge detection and horizontal projection histogram.

1) Vertical edge detection

We perform vertical edge detection on the license plate images obtained using the approach shown in Section II through the computation of horizontal gradient. At each pixel, use the Sobel mask of [-3 0 3;-10 0 10;-3 0 3] to compute the horizontal gradient value. Then, use the Otsu [12] method for binarization to obtain the vertical maps. More detail about Otsu's method for binarization can be found in Subsection B below. The edge pixels are represented by white pixels and other pixels by black pixels. Fig. 2(a) shows the vertical edge pixels on the images in Fig. 1.



(a) (b)

Fig. 2. (a) Vertical edge maps of images in Fig. 1; (b) horizontal projection histograms.

2) Horizontal projection histogram

The use of projection histogram is not a new concept. It, however, contains a new/different idea here is to find the upper and lower bounds of a license plate based on the vertical edges obtained. We perform a horizontal projection to find the top and bottom position of characters. The value of a histogram bin is the sum of the white pixels along a particular line in horizontal direction. When all values along all lines in the horizontal direction are computed, the horizontal projection histogram is obtained. The mean value of the histogram is then used as a threshold to decide the upper and lower bounds. The middle area of which the histogram segment is bigger than the threshold is recorded as the area bounded by the upper and lower bounds. The horizontal projection histograms of the two images shown in Fig. 2(a) are displayed in Fig. 2(b). In order to help the character segmentation process in the next step, we lower the upper bound by two pixels and lift up the lower bound by two pixels.

B. Character segmentation

After the area bounded by the upper and lower bounds of a number plate is found, the areas above the upper bound and below the lower bound are removed. The remaining area without the upper and lower boundaries of the number plate is considered for character segmentation on the number plate. Image binarization and vertical projection are two steps for segmentation.

1) Image binarization

As well known, image binarization is to change grey values of an image into binary values and re-represent the image as a binary image accordingly. Image binarization highlights the pixels of interest and suppresses the background pixels. The simplest way for image binarization is to choose a threshold value, and classify all pixels with values above this threshold as white (255 grey value), and all other pixels as black (0 grey value). Otsu [12] gave an idea to select a good threshold globally. Otsu's method is based on an analysis of the gray scale level histogram of the whole image and selects an optimal threshold for a given image by maximizing a discrimimant criterion, i.e., the separability of the resultant classes in gray levels. Fig. 3 shows the results of binarization on the images in Fig. 1 after cutting the upper and lower boundaries of the license plates.

Fig. 3. Binarized images of the images in Fig. 1 with upper and lower bounds removed

2) Vertical projection histogram

We perform a vertical projection to find the gaps between characters on a license plate. The value of a histogram bin is the sum of the white pixels along a line in vertical direction. When all values along all lines in the vertical direction are computed, the vertical projection histogram is obtained.

Based on the results of vertical projection, each license plate is separated into blocks horizontally by the zero points in the projection histogram. As mentioned in Subsection A of Section III that we lower the upper bound and lift up the lower bound of a license plate before the image regions beyond these two bounds are cut (removed). This is to guarantee that the zero points that separate the characters can be obtained.

Fig. 4 shows the blocks of the images (segmented by vertical white lines) in Fig. 1 obtained from the results of vertical projection after upper and lower bounds removed. It is easy to see that the characters on these two number plates have been well segmented vertically.

Fig. 4. Segmentation of license plates

Note that in Fig. 4, the left and right boundaries of the number plates have also been removed. How these boundaries are removed is beyond this paper but described in another paper in detail.

3) Linking and separation of extracted character segments

Note that in some cases, one character region may have been separated into two segments. One other hand, two characters may have been grouped into one single character segment because they are connected to each other or due to other reasons. Therefore, as a refining segmentation process, any two segments of which the width is 2/3 smaller than the average width of character segments on the same licensa plate are linked (combined) into one segment. Moreover, any segment obtained in the previous step having width 1/3 bigger than the average width is separated into two segments.

For example, as shown in Fig. 5(a), character 'J' was separated into two in the binary image because the real lower bound has been lifted for two much (see Subsection A). Therefore, in the segmentation result, the 'J' region has been separated into two character segments with width 2/3 smaller than the average width (see Fig. 5(b)). We hence need to combine these two segments into one single segment.

(a) (b)

Fig. 5. Character 'J' is separated into two segments by mistake

As another example, Fig. 6(a) shows character 'A' connected to the left boundary of the license plate. Therefore, the left boundary and the region containing 'A' are grouped into one single segment with width 1/3 bigger than the average width after the previous step. We hence need to separate the single segment into two segments. The final segmentation results are displayed in Fig. 6(b).

(a) (b)

Fig. 6. Segmentation of character 'A' from the connected left boundary of the license plate

IV. Experimental Results

We apply the above algorithm to the car license plates on a video footage taken by a CCD camera. Fig. 7 shows some license plate images for our experiments. Altogether, there are 587 license plate images with 3502 characters. All characters are correctly segmented. Fig. 8 shows some of the characters segmented after binarization. The false positives (i.e., non-character areas that are segmented) are mainly due to the false detection of license plates in the step shown in Section 2. There are only 7 out of total 594 license plate detected regions which are non-license plates. This rate is (594-587)/594=1.18%. Therefore, our correct segmentation rate is still very high and about 98.82% even taking into account the rate of wrongly detected license plate regions.

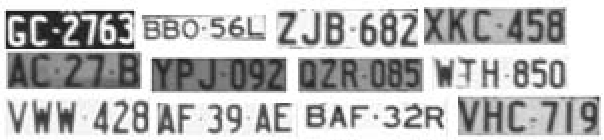


Fig. 7. Some sample license plate images

A performance comparison with various methods is shown in Table I. Our license plate detection rate is 96.4% [13]. Furthermore, our system can detect and recognize a 648*486 license plate image in real-time and within 0.2 seconds on a

PC with Pentium 2.8GHz CPU with a lot of rooms for optimization of our algorithms to be carried out in the future. Comparison of speeds using various LPR systems can be found in [4].

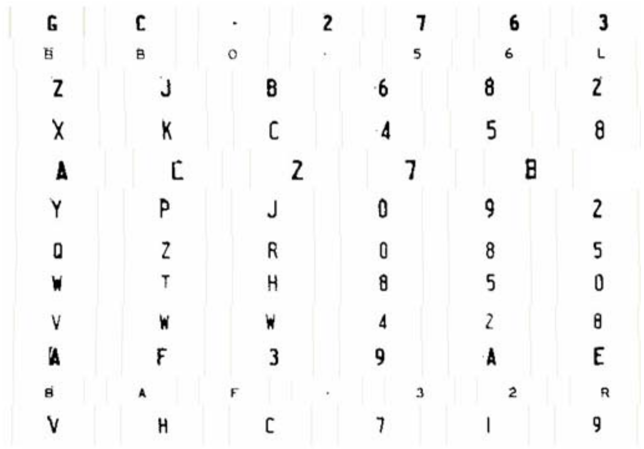


Fig. 8. Some characters segmented

TABLE I
PERFORMANCE COMPARISON [16]

References	Detection Accuracy	Segmentation Accuracy
[14]	96.22%	94.04%
[15]	93.2%	95%
[16]	97.1%	96.4%
Proposed	96.4%	98.82%

V. CONCLUSIONS

In this paper, we have proposed an algorithm for character segmentation of car license plates. This is a crucial work after the detection of license plates and before the character recognition. Various well-known techniques, including AdaBoost algorithm, Hough transform, edge detection, horizontal and vertical projections, and image binarization, are applied to come out with the innovative algorithm in this paper.

The experimental results show that the proposed method is efficient and accurate for character segmentation, and outperforms some very recently developed algorithms as shown in Table I.

Note that the detection, localization, and skew correction of car license plates have been shown in previously published papers (as shown in [3], [11] and [13]) and are beyond the work for segmentation, their detailed description including the AdaBoost algorithm, Hough transform etc. are omitted in this paper.

REFERENCES

[1] http://www.roadtraffic-technology.com/projects/melbourne_citylink/.

[2] http://en.wikipedia.org/wiki/London_congestion_charge.

[3] H. Zhang et al., Learning-Based License Plate Detection in Vehicle Image Database, International Journal of Intelligent Information and Database Systems (IJIIDS), Vol.1, No.2, 2007, pp.228-243.

[4] C. N. E. Anagnostopoulos et al., A License Plate-Recognition Algorithm for Intelligent Transportation System Applications, IEEE Transactions on Intelligent Transportation Systems, Vol.7, No.3, 2006, pp.377-392.

[5] L. W. Jin and J. Z. Qin, Car plate number characters recognition using Gabor orientation features and neural networks, IEEE Int. Conf. Neural Networks & Signal Processing, 2003, pp.1628-1631.

[6] L. Zheng and X. He, Number Plate Recognition Based on Support Vector Machines, IEEE International Conference on Video and Signal Based Surveillance (AVSS), 2006.

[7] Yungang Zhang and Changshui Zhang, A new algorithm for character segmentation of license plate, IEEE Intelligent Vehicles Symposium, 2003, pp.106-109.

[8] Jian Zhang, Xiaoping Fan and Cailun Huang, Research on Characters Segmentation and Characters Recognition in Intelligent License Plate Recognition System, Proceedings of the 25th Chinese Control Conference, 2006, pp.1753-1755.

[9] Xiaodan Jia, et al., A Novel Algorithm for Character Segmentation of Degraded License Plate Based on Prior Knowledge, Proceedings of the IEEE International Conference on Automation and Logistics, 2007, pp.249-253.

[10] P. Viola and M.J. Jones, Robust Real-Time Face Detection. International Journal of Computer Vision, 2004. 57(2): pp. 137-154.

[11] Wu Dong, Xiangjian He, and Tom Hintz, A Robust Hough Transform Based Method for Direction Detection, International Conference on Image Processing, Computer Vision and Pattern Recognition (IPCV), 2007, pp.439-443.

[12] N. Otsu, A threshold selection method from gray-level histograms, IEEE Transactions on Systems, Man, and Cybernetics, vol. 9, 1979, pp. 62-66.

[13] X. He et al., Combining Global and Local Features for Detection of License Plates in Video, Proceedings of Image and Vision Computing New Zealand, 2007, pp.288-293.