Proceedings of the Federated Conference on
Computer Science and Information Systems pp. 149–154
ISBN 978-83-60810-51-4

A Real-Time License Plate Localization Method Based on Vertical Edge Analysis

Peter Tarabek
University of Zilina, Department of Transportation Networks, Zilina, Slovakia
Email: peter.tarabek@fri.uniza.sk

***Abstract*—License plate localization is the most important part of the license plate recognition system. Ability to correctly detect license plate under different conditions directly affects overall recognition system accuracy. In this paper a real-time license plate localization method is proposed. First, vertical edges are detected from the image and binarized. Then, license plate candidates are extracted by the two-stage detection process. In this process a sliding-window technique is used to mark all windows which satisfied edge density conditions. Edge density conditions are computed on integral edge image allowing us to significantly increase the processing speed of the method. To better distinguish between license plates and complex backgrounds, the edge analysis is performed to remove specific edges. Finally, false candidates are filtered out based on geometrical and textural properties. The proposed method can detect multiple license plates with different sizes in a complex background. The experimental results confirm robustness and ability to localize license plates in real-time. On the database of 501 images our method correctly localizes 97.4% of license plates.**

I. Introduction

LICENSE plate recognition (LPR) is used as a core module for many intelligent transportation systems such as automated traffic surveillance and tracking systems or electronic payment systems. Roughly, an LPR can be divided into three stages: 1) license plate localization (LPL); 2) character segmentation; and 3) character recognition. Among these, the LPL is the most challenging stage. For the overall system to be accurate, the LPL has to locate license plate (LP) correctly in different conditions, while it should be fast enough to meet the needs of intelligent transformation system.

Variability of the environment and conditions in which the image is captured as well as variations in LP texture are the main challenges for LPL systems. The image can be captured in various illumination conditions and may contain one or more LPs in various locations and sizes. Texture of the LP can vary due to shadows, dirt and plate types. Also a background may include trees, buildings, fences, floors, cars bumpers and headlights, etc., containing patterns similar to LP.

This work has been supported by grants VEGA 1/0296/12, APVV-0760-11 and FVG 2012.

Usually the LPL consists of two phases: candidate detection and candidate verification. Tan and Chen [1] proposed candidate verification method based on adaptive binarization. Gradually, three different binarization methods are used to extract different plates. In each step conditions based on auto-correlation, projection and character position are checked to detect LP. Wu et al. [2] proposed morphology-based method, which can extract LPs from low resolution video. The bottom-hat operator, which can enhance the edge and suppress the homogeneous region, is applied on the image. Since the bottom-hat operator can enhance various kinds of texture, morphological gradient is used to distinguish denser regions. Connected components are recognized in closed and binarized image and each component is verified based on geometrical and textural properties. During the license plate detection it is assumed that a preprocessing module exists to detect the region of moving vehicle so the search is done only in this limited area. Another assumption is that the distance between the video camera and the observed vehicle falls within a known range. In Mahini et al. [3] regions with many vertical edges, gray looking regions and light background regions are merged together to find plate candidates. Morphological operations and edge detection are used to find these regions. After the candidates are found, verification process is applied. To be able to detect different LP sizes, the image is gradually resized and localization process is applied until the LP is detected. This process ensures localization of only one LP per image. Additionally, characters in LP have to be darker than plate background. A hybrid license plate extraction method based on edge statistics and morphology is proposed in [4]. First, the candidates are located hierarchically based on edge statistics. Than, if no candidate region is found, morphology-based location is applied. Although the reported average accuracy is 99.6%, the images were acquired from a fixed distance. Also the number of false candidates (false positives), which has not been directly mentioned in this paper could be high. Guo and Liu [5] proposed LPL with feedback self-learning. Edge detection, texture property, aspect ratio and color switch property are used to find the LP. However, this method can only localize LP with fixed size. In [6] vertical edges are extracted and background curves and noise are removed. Then, the search window is convolved with image to find the LP. The method works best if all LPs in images have the same size. In [7] horizontal gradient detection following by mean filter are used to obtain LP regions. In the next phase, non-LP regions are darken by morphological operations. Then, the image is binarized and processed by connected component analysis to identify LP candidates. The method is design to localize one LP in image. This is due to the statistical measures which aim to identify the most probable LP candidate. Jia et al. [8] utilized mean-shift color segmentation to find candidate regions. They defined three features in order to separate the plate area from other candidate regions. Based on their statistical analysis, the LP can be characterized by a unique combination of three features: rectangularity, aspect ratio and edge density. The algorithm can accurately extract LP area but the color of the LP needs to be different from the vehicle color. Moreover, the method is computationally intensive even for small color images (324x243). A Survey aimed to LPR from still images and video sequences was presented by Anagnostopoulos et al. [9].

While it is not unusual for current methods to have reported accuracy above 90%, in most cases they work well only under the specific conditions. Some of these restrictions include fixed size LP detection and localization of one LP per image. They also tend to have high false detection rate in complex backgrounds and high computational complexity.

In this paper, new method for LPL which aims to overcome these issues is proposed. It consists of three steps: vertical edge detection, two-stage candidate detection and candidate verification. The novelty of the method lies in vertical edge analysis which allows us to better distinguish LPs from complex backgrounds and in two-stage candidate detection process and usage of integral edge image which allow us to process 640x480 images in real-time. The method can detect multiple LP with different sizes in a complex background. Experiments performed on data sets from LPR database presented in [9] and its modification used in [11] confirm accuracy and low computational complexity of our method.

The rest of this paper is organized as follows. The proposed method for LPL is presented in Section II. The experimental results are shown in section III. Finally, the conclusions are drawn in section IV.

II. License plate localization

Ability to process images in different illumination conditions plays an import role in LPL systems. Typically, the LP contains several letters and numerals which color significantly differs from their background. This means that the plate area is rich in edge and texture information. Edges are robust to illumination changes and therefore we based our method on vertical edges. Fig. 1 shows overview of the proposed method.

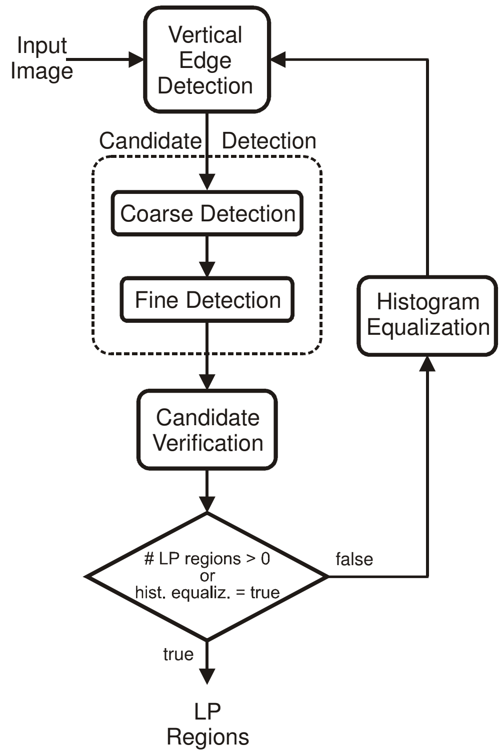


Fig. 1 Overview of the proposed LPL

First, the vertical edges are detected and then candidate detection is performed. Candidate detection process is divided into two steps: coarse and fine detection. The role of the coarse detection is to quickly remove easily distinguishable background regions from the image. The remaining regions are further processed by the fine detection to localize LP candidates. Subsequently, the candidates are filtered in candidate verification step using geometric and textural properties. The remaining candidates are declared as LPs. In the case that no candidate is left, the input image is processed by the histogram equalization and the whole process is repeated.

Estimation of parameters was done using a training set which contains 40% of all images from large database described in section III.

A. *Vertical Edge Detection*

We use vertical edge detection because it preserves majority of edge information in the plate area while it removes lots of horizontal edges around the LP. This makes further localization process easier. Fig. 2 shows an example of vertical and horizontal edge detection. There are many methods suitable for vertical edge detection. We have chosen Sobel operator (shown in Fig. 3) because it is computationally inexpensive and reasonably robust to noise. After the edges are detected, a binary edge map is obtained by adaptive binarization.

(a) (b)

Fig. 2 Vertical (a) and horizontal (b) edge detection

$$\begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}$$

Fig. 3 Sobel operator for vertical edge detection

B. Candidate Detection

To find LP candidates a sliding window approach is adopted. At the beginning window of minimal size is moved through the binary edge map. Each window is separately evaluated and all windows satisfying conditions are saved for further processing. After the whole image is processed the size of the window is increased and the evaluation process continues until the maximum window size is reached.

Sliding window detection has one fundamental weakness: it is very computational expensive to search and evaluate all possible window locations. To deal with this issue the number of evaluations is reduced using the translation and size steps which define how fast the window is moved and resized. These steps need to be chosen carefully because they represent a trade off between the speed and the localization robustness of the LPL method. Another important aspect of candidate detection procedure is an evaluation function which needs to be fast and accurate.

The proposed evaluation function i.e. the binary classifier is based on edge density features and hierarchically structured conditions. These conditions are gradually controlled one by one and if any of the conditions is not satisfied the window is rejected meaning that the remaining conditions and features are not computed at all. The edge density can be computed by following equation:

$$ed = \frac{1}{w \times h} \sum_{x=1}^{w} \sum_{y=1}^{h} e(x, y) \tag{1}$$

Where w and h are the width and height of the image and $e(x,y)$ is the binary edge map. Because the edge density is computed as rectangular sum in the edge density map the computation can be greatly reduced using the concept of integral image introduced in [10]. Similarly to this work, the integral edge image can be computed in one pass over the binary edge map using the following terms:

$$r_{sum}(x, y) = r_{sum}(x, y-1) + e(x, y) \tag{2}$$

$$I(x, y) = I(x-1, y) + r_{sum}(x, y) \tag{3}$$

Where r_{sum} is the cumulative row sum, $e(x,y)$ is the binary edge map, $I(x,y)$ is the integral edge image, $r_{sum}(x,-1) = 0$ and $I(-1,y) = 0$. Once the integral edge image is computed the edge density can be computed in four array references:

$$\begin{aligned} ed_I = I(x_{\max}, y_{\max}) - I(x_{\max}, y_{\min}) \\ - I(x_{\min}, y_{\max}) + I(x_{\min}, y_{\min}) \end{aligned} \tag{4}$$

Where I is integral edge image, X_{min}, X_{max}, Y_{min} and Y_{max} are boundary coordinates of window.

There is one important property of the edge density which needs to be considered. As can be seen in Fig. 4, the edge density of LPs is not constant and it decreases with increasing plate size. To be able to localize LP with high diversity of sizes, adaptive threshold values based on the actual size of the sliding window need to be used by classifier.

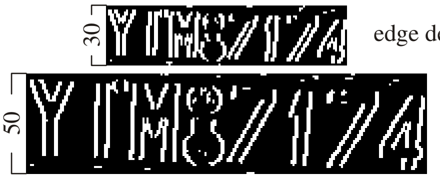


Fig. 4 Example of edge density of the LP with different heights

C. Coarse Candidate Detection

The candidate detection is divided into two sequential steps: coarse and fine detection. The purpose of the first step is to quickly reject majority of background regions while detecting all LP regions. During the coarse detection the edge density, computed using the integral edge image, in the window with fixed aspect ratio is controlled. If the density is higher than a threshold value (0.15 in our experiments), the window is saved. The parameters of the sliding window were: horizontal and vertical translation step = 15% of the current width and height of the window, size step = 40% of its current size, minimum window height = 20, maximum window height = 40, and aspect ratio = 4.0. The result of the candidate detection is shown in Fig. 5(c).

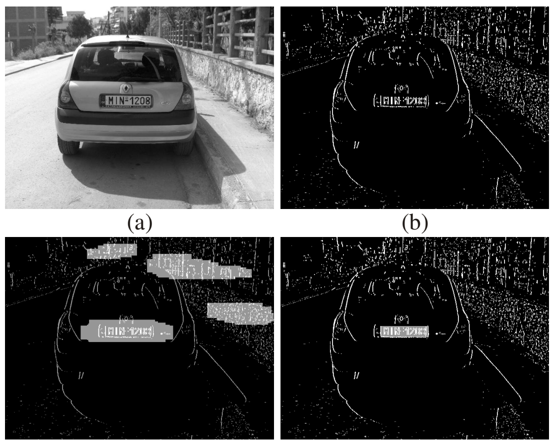


Fig. 5 Result of candidate detection: (a) input image, (b) binary edge map, (c) coarse detection, (d) fine detection

D. Fine Candidate Detection

In the second step, the coarse candidates are further examined to distinguish background regions with similar edge density from the LP. LP regions have some unique edge properties which are used to identify their patterns. They are:

a) Certain edge density (usually higher than background objects).
b) Relatively uniform distribution of edges.
c) Relatively high edges compared to plate height.
d) Edge height is smaller than plate height.

The first two properties can be easily checked and are often used in different edge and morphology based LPL methods. The problem of properties (c)-(d) is that it can be computationally expensive to include them into the detection process. For this reason they are rarely used or they are used only in some very limited way e.g. only in verification step or to detect LPs with known size.

To utilize these properties an edge analysis procedure is proposed. It removes all edge components not satisfying following condition:

$$0.1 \times W_H \leq C_H \leq W_H \quad (5)$$

Where W_H and C_H are the height of the current window and the height of the edge component respectively. Because the analysis depends on actual window size it has to be executed every time the window height has changed. To improve efficiency of the procedure, a list of edge components and temporal edge map are created. At the beginning all edge components are added to the list. After the window height is set, edge components are analyzed to permanently remove small edges from both the list and the original edge map. The edges larger than the window height are not removed from the list. In the next iteration, when the window height will be increased, these edges have to be re-evaluated. For this reason they are preserved in the list and are removed only from the temporal edge map. After the analysis is performed the integral edge image I_2 of the temporal edge map is computed and is used until the height of the sliding window is changed. This means that there are two integral edge images used to compute edge density features. I_1 computed from the edge map obtained directly from vertical edge detection step and I_2 computed from the temporal edge map obtained from the edge analysis procedure.

The height of the window is also used to adaptively choose following threshold values: minimum edge density in I_2 (ed_{min}) and maximum difference of densities between I_1 and I_2 ($ed_{diffmax}$). These threshold values were estimated for the minimal and maximal LP height using the training set. Based on these values the specific threshold value for the current window is computed as follows:

$$ed_{min} = T_{min} - \left(\left(\frac{T_{max} - T_{min}}{H_{max} - H_{min}} \right) \times \left(H_c - H_{min} \right) \right) \quad (6)$$

Where T_{min} and T_{max} are estimated threshold values for minimal and maximal LP height. H_{min}, H_{max} and H_c are minimal, maximal and current height of the sliding window. For the databases used in our experiments T_{min} = 0.101, T_{max} = 0.063, H_{min} = 20, H_{max} = 100.

Similarly $ed_{diffmax}$ is computed, where T_{min} = 0.021 and T_{max} = 0.026.

Three types of conditions are proposed to detect fine candidate windows: overall edge density conditions, uniformity conditions, and accuracy conditions.

Overall edge density conditions

The evaluation starts by comparing the window overall edge density. The edge densities in I_1 and I_2 are computed and the window is rejected if it does not fulfill following two constraints:

$$ed_{min} \leq ed_{I2} \quad (7)$$

$$ed_{I1} - ed_{I2} \leq ed_{diffmax} \quad (8)$$

Where ed_{I1} and ed_{I2} are the edge density of the window in I_1 and I_2 respectively.

Uniformity conditions

If the window was not rejected by previous conditions, the uniformity of edge distribution is tested. The window is divided in four equal subwindows as shown in Fig. 6(a). The window is preserved if the edge density of each sub window is higher than ed_{min}. This condition will quickly remove windows with non-uniform edge distribution.

Preserved window is further analyzed. It is divided to 20 vertical and 10 horizontal subwindows (Fig. 6(b)) and the number of subwindows with the edge density higher than ed_{min} is computed. If it is ≤ 21 (70% of all subwidnows), the window is rejected.

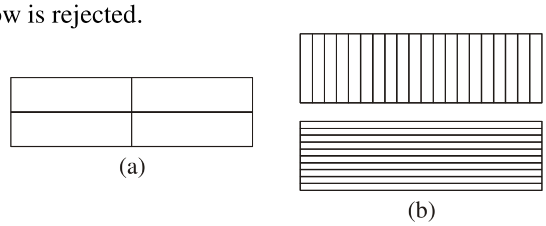
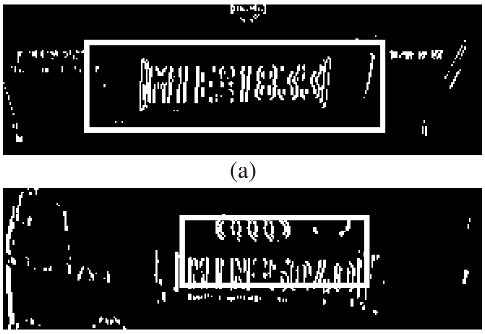


Fig. 6 Subwindows

Accuracy conditions

These conditions aim to discard windows which include part or whole LP, but are either too large (Fig. 7(a)) or contain other background objects (Fig. 7(b)).

By discarding these windows an accuracy of location is improved and percentage of false negative in verification phase is lowered. First, the edge density of the window border is tested, and then horizontal and vertical sliding sub windows are used to detect the space between LP and other objects. Examples of rejected windows are shown in Fig.7.



(a)

(b)

Fig. 7 Examples of inaccurate candidate windows (a) window is too large, (b) window contains other objects

E. Candidate Verification

After the detection process, candidates are extracted using the connected component analysis. Three conditions based on geometrical properties and horizontal projection are employed. If the candidate satisfies all of these conditions, it is declared as LP.

Because the LPs have relatively uniform size and shape, each candidate needs to satisfy following conditions:

$$t_1 \leq \frac{B_W}{B_H} \leq t_2 \tag{9}$$

$$\frac{B_W \times B_H}{AREA} \leq t_3 \tag{10}$$

where B_W and B_H are the width and height of candidate bounding box, t_1, t_2 and t_3 are threshold values and *AREA* is the number of active pixels in candidate area. In our experiments the threshold values were 2.2, 6.2 and 1.2.

To remove false candidates with similar pattern to the LP, the horizontal projection of candidate is computed. The number of peaks *NP* higher than the threshold (0.2*height of the candidate) should be:

$$t_4 \leq NP \leq t_5 \tag{11}$$

where t_4 and t_5 are threshold values (9 and 40 in our experiments).

III. Experimental results

Experiments were performed on two databases denoted as small and large database. Small database consists of Greek LPs with total number of 335 images, available in [11]. Large database was created by adding additional images from the LPR database presented in [9]. We use images from following four data sets: Day (color images large sample), Day (close view), Day (with shadows), Shadows in plate. According to the authors the database includes image and video data of Greek LPs collected and grouped according to criteria such as type and color of plates, illumination conditions, various angles of vision, indoor or outdoor images. Together, large database consists of 501 unique images with minimum and maximum LP height 20 and 99 respectively.

We use small database mainly for comparison of the proposed method with the work of Mendes et al. [7]. This work was chosen for two reasons. First, their method yield very good results which are confirmed by comparison with other state of the art methods. Second, they introduced a methodology to statistically evaluate quality of the located LPs, and they also provide annotated database (small database) which allows us to precisely recreate the experiment. To evaluate the quality of the localization, the located area *la* and the excessive area *ea* are defined as follows:

$$la = \frac{area(r_{char} \cap r_{met})}{area(r_{char})} \tag{12}$$

$$ea = \frac{area(\overline{r_{vlp}} \cap r_{met})}{area(r_{vlp})} \tag{13}$$

Where r_{char} is minimum bounding box that includes all LP characters, r_{vlp} is minimum bounding box that includes the entire LP, r_{met} is the LP region found by the method and *area*() is a function which returns area (in pixels) of a given region.

The results obtained with both databases are shown in Table 1. Usually the methods are evaluated using the naïve localization rate which not considers the quality of the localized LPs. In small database our method was able to reach 97.97% which is slightly less than the Mendes method. However, the optimum location rate, location error and the mean number of candidates are slightly better for our method. Examples of localized LPs are shown in Fig. 8. Although the results of proposed method and the Mendes method are very similar, our method is able to detect multiple LPs in image. This results from the design itself but was also confirmed by the experiment where groups of four images with similar lightning conditions were randomly chosen from large database and combined together to form images with multiple LPs. The location results in these images were approximately the same as the localization results in original single images.

The method was implemented mainly in C# (some of the routines were implemented in C++) and tested on an Intel Core 2 Duo 2.53GHz with 2GB of RAM. The average processing time of one 640x480 image was 36 ms what meets the time required for real-time processing. We believe that the code can be further optimized to process larger images.

IV. Conclusions

A fast and robust method for license plate localization which uses edge information is presented in this paper. The proposed approach can be divided into three steps: vertical edge detection, candidate detection and candidate verification. Edge density conditions are used to find LP

Table I.
The Experimental Results

	optimum location *la* > 85% and *ea* < 100%	**excessive location** *la* > 85% and *ea* >= 100%	**location error** *la* <= 85% and *ea* < 100%	**"naive" location** *la* > 0	**mean number of candidates**
small database					
Mendes et al. [7]	96.52%	0.87%	2.61%	99.13%	1.53
Proposed method	97.68%	0.00%	2.32%	97.97%	1.14
large database					
Proposed method	97.41%	0.80%	1.79%	98.60%	1.14

candidates. The threshold values are adaptively chosen depending on the height of the sliding window. This allows us to detect LPs with wide range of sizes. The verification step removes some of the false candidates which have similar pattern to LP area. The method can process 640x480 images in real-time and can also localize multiple LPs in one image. Current approaches usually reduce false detection by utilizing a prior knowledge about position, size and number of LPs in the image. This kind of information can be easily adopted by our approach as well to further reduce number of false candidates. Although the proposed method was calibrated and tested on Greek LPs, we believe that this concept can be modified to localize also other types of LPs and that the proposed method can find applications in various intelligent transportation systems.

(a)
(b)
(c)

Fig. 8 Examples of localized LPs: (a) complex background with small LP, (b) blurred image, (c) LP with shadow

ACKNOWLEDGMENT

This work has been supported by grants VEGA 1/0296/12, APVV-0760-11 and FVG 2012.

REFERENCES

[1] H.-C. Tan, and H. Chen, "A novel car plate verification with adaptive binarization method," in *Proc. of the 7th ICMLC,* 2008, pp. 12-15.

[2] P. Wu, H.-H. Chen, R.-J. Wu, and D.-F. Shen, "License plate extraction in low resolution video," in *Proc. 18th ICPR*, vol. 1, 2006, pp. 824-827.

[3] H. Mahini, S. Kasaei, F. Dorri, and F. Dorri, "An efficient features–based license plate localization method," in *Proc. 18th ICPR*, vol. 2, 2006, pp. 841-844.

[4] B. Hongliang, and L. Changping, "A hybrid license plate extraction method based on edge statistics and morphology," in *Proc. 17th ICPR*, 2004, pp 831-834.

[5] J.-M. Guo, and Y.-F. Liu, "License plate localization and character segmentation with feedback self-learning and hybrid binarization techniques," *IEEE Trans. Veh. Technol.*, vol. 57, no. 3, pp. 1417–1424, 2008.

[6] D. Zheng, Y. Zhao, and J. Wang, "An efficient method of license plate location," *Pattern Recognit. Lett.*, vol. 26, no 15, pp. 2431-2438, 2005.

[7] P.R. Mendes, J.M.R. Neves, A.I. Tavares, and D. Menotti, "Towards an automatic vehicle access control system: License plate location," in *IEEE International Conference on Systems, Man, and Cybernetics (SMC),* 2011, pp. 2916-2921.

[8] W. Jia, H. Zhang, X. He, and M. Piccardi, "Mean shift for accurate license plate localization," in *Proc. 8th Int. IEEE Conf. Intell. Transp. Syst.*, 2005, pp. 566–571.

[9] C. E. Anagnostopoulos, I. E. Anagnostopoulos, and I. D. Psoroulas, V. Loumos, and E Kayafas, "License plate recognition from still images and video sequences: A survey," *IEEE Transactions on Intelligent Transportation System*, vol. 9, no. 3, pp. 377–391, 2008.

[10] P. Viola, and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. CVPR*, 2001, pp. 511–518.

[11] P. R. Mendes-J´unior, D. Menotti, J. M. R. Neves, and A. I. Tavares, "Vehicle license plate location (VLPL) algorithms," [Online]. Available: https://github.com/pedrormjunior/vlpl.