Real-time license plate localization based on a new scale and rotation invariant texture descriptor

Chu-Duc Nguyen, Mohsen Ardabilian, Liming Chen

Real-time license plate localization based on a new scale and rotation invariant texture descriptor

Chu-Duc Nguyen, Mohsen Ardabilian and Liming Chen

Lyon Research Center for Images and Intelligent Information Systems, UMR 5205 CNRS
Department of Mathematics and Computer Science, Centrale Lyon
36 avenu Guy-de-Collongue, 69134 Ecully, France
chu-duc.nguyen@ec-lyon.fr, mohsen.ardabilian@ec-lyon.fr, liming.chen@ec-lyon.fr

Abstract— In this paper, we present a real-time and robust license plate localization method for traffic control applications. According to our approach, edge content of gray-scale image is approximated using line segments features by means of a local connective Hough transform. Then a new, scale and rotation invariant, texture descriptor which describes the regularity, similarity, directionality and alignment is proposed for grouping lines segments into potential license plates. After a line-based slope estimation and correction, false candidates are eliminated by using geometrical and statistical constraints. Proposed method has been integrated in a optimal license plate localization system. Evaluation is conducted on two image databases which were taken from real scene under various configurations and variability. The result shows that our method is real-time, robust to illumination condition and viewpoint changes.

Index Terms— License plate localization, Hough transform, scale and rotation invariant texture descriptor, slop estimation, motion analysis.

I. INTRODUCTION

License Plate Recognition (LPR) system has played a very important role in intelligent traffic control and security management, such as highway ticketing, car park entrance and exit management [19]. Usually, an LPR system consists of three main parts: license plate localization, character segmentation, and character recognition. Among them, license plate localization (LPL) is considered the most crucial stage because a high accuracy and real time segmentation and recognition can be performed only if the license plates are correctly localized [18][11].

In real-life context, the LPL has to confront some difficulties which result from uncontrolled imaging conditions such as complex scene, bad weather condition, low contrast, blurring and viewpoint changes [11][7]. Even though many researches focus on LPL, we are far from a solved problem.

In this paper, a real-time and robust LPL method for traffic control applications will be presented. As the main contribution, we propose a new, scale and rotation invariant, texture descriptor which describes the regularity, similarity, directionality and alignment of line region for localizing LP. First, edge content of gray-scale image is approximated using line feature extracted by a local Connective Hough Transform. Then line segments are grouped into potential LP by using our texture descriptor. The new texture

descriptor and the real-time extraction approach guarantee some remarkable properties as insensitivity to illumination, insensitivity to geometric change induced of the change of viewpoint (scaling, orientation and deformation) and could be applied on gray-scale images without binarisation. For evaluation purpose, proposed method has been implemented in a complete license plate localization system.

The rest of the paper is organized as follows. The section II introduces a state of the art of LPL approaches. In section III, our new line segments detection and line-based texture descriptor are presented. In section IV, we propose a global traffic control system architecture in which our approach is optimally involved. LP databases are also described inthe section V with experimentation and performance analysis. Finally, conclusions and perspectives come in the last section.

II. RELATED WORKS

LPL process could be divided into two consecutive phases: all potential LP regions detection and LP verification to eliminate false alarms. Researchers' answer for each phase differs about the feature type used. In the light of this observation, LPL methods can be categorized into three global types: region-based, contrast-based and hybrid methods.

Region-based methods use the properties of the color in LP region, or the character information of the alphanumerical components inside LP. Color-based methods extract LP's background pixels by using some predefined colors [19] or color image segmentation [9]. CC-based methods [12][10] use a bottom-up approach by grouping small components into successively larger components until all regions are identified in the image. After a non-like character filtering stage, a geometrical analysis is needed to merge the text components using the spatial arrangement of the components.

Contrast-based methods are founded on the contrast between the plate boundary and overlapping region in the vehicle or the high contrasted characters in comparison with the rest of the plate region. This category of methods can be categorized into three sub-types: straight-line-based, morphology-based and edge-based. Straight-line-based methods detect lines in binarized edge image and then group two pairs of overlapping parallel lines in the horizontal direction and the vertical direction to form potential LP boundaries

[5][17]. Morphology-based methods [18][3] use morphology operators to enhance the local regions which are rich in contrast information (LP region is belong to this type) and reduce the other regions. Then, a thresholding step removes regions with low edge density. Edge-based methods occupy the majority of existing methods, generally, thanks to their high localization capacity in various conditions and their low computation time. This sub-category of approaches exploit the edge information of characters inside LP, especially the vertical strokes properties [8][11][16][4].

Hybrid methods are developed to face the large diversity of LP color and contrast information where a single type approach suffers from completeness. Heo et al. [6] propose a double chance scheme using line grouping and edge density methods. Porikli [14] proposes a new descriptor based on covariance matrix of spatial features and appearance features of image regions. This descriptor has low dimensionality, the same size for any regions and is invariant to in-plane rotation. Zhang [20] uses a cascade classifier of six layers by extracting gradient features and adaboosting of Haar-like features on the vertical edges image. Hongliang [7] proposes a hybrid scheme of scanning-line technique on multiple thresholded vertical edge images and morphology-based localization on vertical edges density map.

For finding an efficient solution to LPL problem, the pros and cons of existing approaches have been analyzed. Color is very useful when the lighting condition is good, but color information is not stable when the illumination changed. CC-based methods are robust to viewpoint and illumination conditions, but interference characters cause high false positive rate. Methods based on plat boundary detection can solve the scale and orientation problems; however the plate boundary is not always clear in real scenes. Morphology-based methods are robust to noise and broken character strokes, but its slow operation makes it rarely used in real-time system. Hybrid methods with learning-based techniques are very efficient in many scenes nevertheless they are time consuming at the learning stage, needs a large and diversified image database and are database dependent. Edge-based methods are widely used for the advantage of plate candidates under different lighting conditions efficiently and fast. The main inconvenient of this strategy is its noise sensitivity. Hence, robust constraints are vital in the LP verification stage.

This analysis leads us to an approach combining edge and line segments information for localizing LPs. This approach exploits optimally vertical strokes features of characters inside LP. Firstly, only intensity component has been used to guarantee the robustness to illumination change. Secondly, a fast approximation of straight edge content by line segments, without binarization, decreases dramatically the overall computation time. Thirdly, a new, invariant scale and rotation invariant, texture descriptor is proposed for grouping line segments into potential LPs, which gives us a reliable LP localization even when viewpoint changes.

III. LINE SEGMENTS BASED TEXTURE DESCRIPTOR

First of all, let us consider the idea of our line segments based texture descriptor (LSTD). It is a well-known fact that color information is not stable when the illumination changes, so only intensity component will be exploited. In this gray-scale image, straight edge content can be approximated by line segments. By using these segments, we want to build a descriptor which represents a perceptual characterization of texture, similar to a human characterization, in terms of regularity, similarity, directionality and alignment. A similar idea of perceptual texture extraction for artificial text detection has been used in [2]. We remind here that regularity and directionality features are specified in MPEG-7 and more specifically in Texture Browsing Descriptor [13].

A. Descriptive of line segments based features
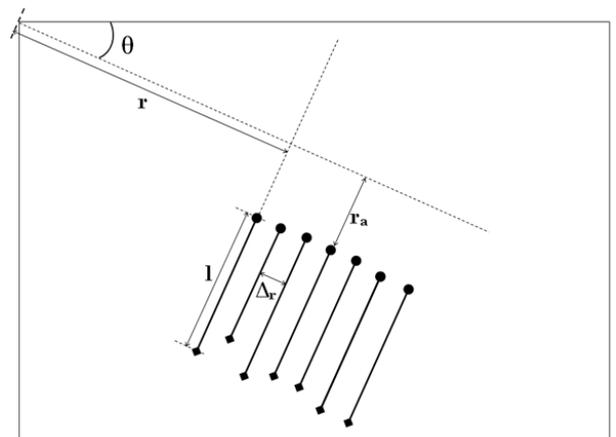

Fig. 1.   Line segments based texture representation

The syntax of our LSTD, which can be deduced from Fig. 1, is the following:

$$LSTD = [\Delta_r, l, \theta, r_a] \qquad (1)$$

where each feature represents a characteristic of line region:

- *Regularity* is represented by Δ_r, vector of distances between two adjacent line segments which have the same orientation.
- *Similarity* is represented by l, vector of lengths of line segments.
- *Directionality* is represented by θ, vector of angles between the horizontal axis and the perpendicular line of line segments.
- *Alignment* is represented by r_a, vector of distances between extreme of line segments and the perpendicular line of line segments.

One of the most striking aspects of our LSTD is the invariance to rotation and scale changes. Let us firstly consider the case in which the angle θ of line segments has been changed an amount of Δ_θ. The vector of features becomes:

$$LSTD^{'} = [\Delta_r, l, \theta^{'}, r_a] = [\Delta_r, l, \theta + \Delta_\theta, r_a] \qquad (2)$$

Every θ_i are added a same value of rotation, therefore the new line segments group had the same directionality according to the initial group.

On the other hand, take the case of image scale change by factor α. Region features becomes:

$$LSTD'' = [\Delta_r', l', \theta, r_a'] = [\alpha.\Delta_r, \alpha.l, \theta, \alpha.r_a] \quad (3)$$

All of r_i, l_i and r_{ai} are modified with the same ratio α, while θ_i stays intact. On that account the scaled line segments group has the same LSTD features with respect to the group in original image.

B. Feature extraction based on local connective Hough transform

Our feature extraction is based on line detection by using local Hough transform combined with small connected line segments suppression. The algorithm is based on an original fast connective Hough transform in [1]. The issue at stake here is the minimization of noise's effect caused by isolated pixels. As for rapidity, small segments suppression also reduces the number of pixels to be considered in line detection stage. Additionally, lookup-table technique is also used to store and involve values of trigonometric functions, which are the same for every local sweeping using Hough transform. As a contribution, our local sweeping scheme can reduce significantly the computation cost of classical Hough transform.
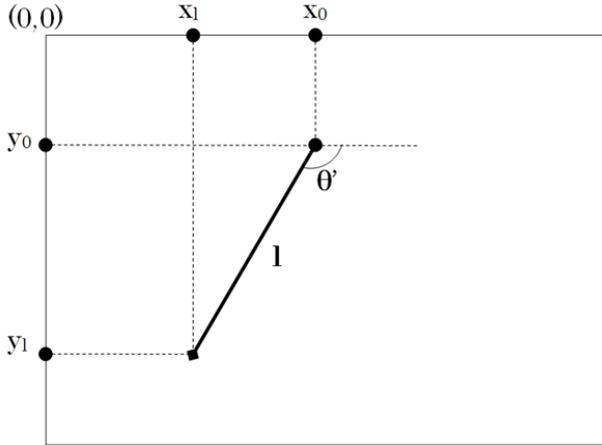

Fig. 2. Line representation based on local features

1) Line segment detection in binarized image: A simple way to get all pixels of a line segments is to scan sequentially the image from top to bottom and from left to right. For each foreground pixel, every local angle θ' which is between 0 and π will be considered (Fig. 2). In each direction given by θ', line segment is detected by searching a set of connected collinear pixels. Finally, the longest line segment is stored and removed from the image. All other small segments whose length is lower than a threshold are also removed. The similar processus has been used in [15] for a line-based cityscape classification.

The sequential search continues until sweeping the whole image. The detection of segment direction is the most costly step in line detection algorithm, so requires a particular attention. In the discrete image space I, pixels of line segment LS having a local direction θ', length l and beginning coordinates (x_0, y_0) can be represented by:

$$LS_{\theta'} = I(x_0 + \lfloor i.cos(\theta') \rfloor, y_0 + \lfloor i.sin(\theta') \rfloor) \quad (4)$$

where $1 \leq i \leq l$, $0 \leq \theta' \leq \pi$ and $\lfloor . \rfloor$ denotes entire part of a number. In order to improve performances and avoid call of trigonometric functions inside our algorithm, we compute two transformation matrixes in the initialization step. The first one stocks values of $\lfloor i.cos(\theta') \rfloor$, the second one stocks values of $\lfloor i.sin(\theta') \rfloor$ for $1 \leq i \leq l_{max}$, $0 \leq \theta' \leq \pi$, where l_{max} is the maximal length of line segment that we want to detect. At the end of a local line detection cycle, all segments
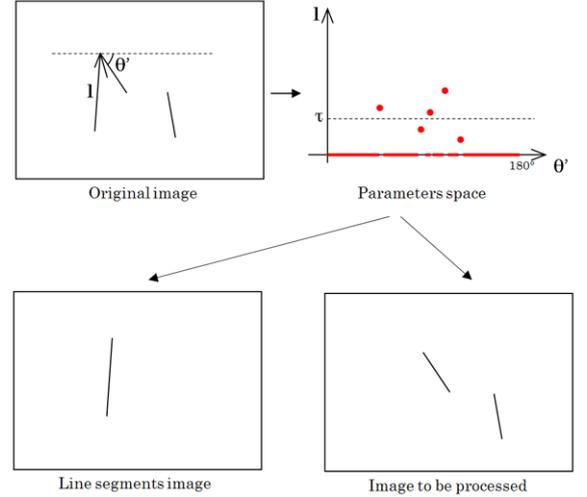

Fig. 3. A cycle of line segments detection for a given pixel

having length lower than a threshold τ will be removed from image space. Fig. 3 serves to illustrate the idea of this step.

2) Line segment detection in gray-scale image: There can be no doubt that in gray-scale image, line segments are often imperfect, because of small variances in intensity, in space or in orientation. So, we introduce tree tolerant parameters:

- *Tolerant in intensity*: gray-scale value of pixels in a given direction can be different of τ_{gs} levels.
- *Tolerant in space*: in a given direction, τ_d neighbors of current pixel will be considered.
- *Tolerant in orientation*: in a given direction θ', neighbors pixels in directions $\theta' \pm \tau_\theta$ will be also considered.

This consideration of neighborhood makes our method able to detect imperfect segment as the case of an edge image.

3) Application to LSTD features extraction: The above section shows how our method can detect rapidly and robustly line segments in gray-scale image. Therefore, in order to use detected line segments to extract LBTD features, some transformations are needed (Fig. 4).
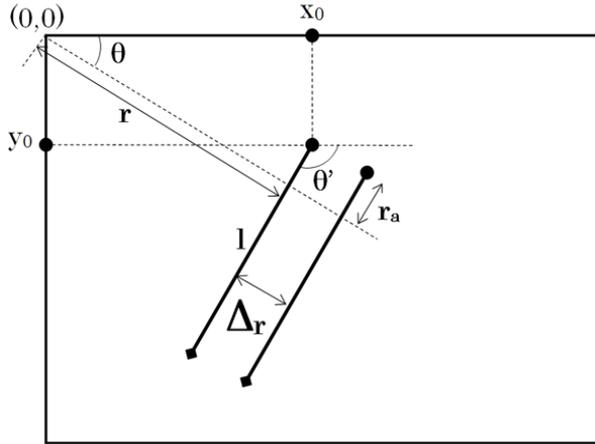
Fig. 4.   LBTD features extraction

- *Extraction of regularity features*: firstly, the distance r between the origin and a given line segment is calculated from its extreme coordinates (x_0, y_0) and its local orientation θ':

$$r = |x_0.sin\theta' - y_0.cos\theta'| \qquad (5)$$

Secondly, Δ_{ri} between every parallel line segments will be readily deduced.

- *Extraction of similarity features*: which are lengths l of line segments.
- *Extraction of directionality features*: for $\theta' \in [0, \pi]$, two cases are presented for calculating θ :
  If $\theta' \in [0, \pi/2]$ then $\theta = \theta' + \pi/2$
  If $\theta' \in [\pi/2, \pi]$ then $\theta = \theta' - \pi/2$
- *Extraction of alignment features*: distances r_a between the extreme of line segments and the perpendicular line of line segments can be calculated directly from extreme coordinates (x_0, y_0) and local orientation θ' as:

$$r_a = |x_0.cos\theta' + y_0.sin\theta'| \qquad (6)$$

IV. LICENSE PLATE LOCALIZATION SYSTEM

In this section, we describe the main steps of the license plate localization system by using LBTD. It is known that a license plate is a pattern composed of several characters that have high distinctive intensities in LP background. The second observation is that characters inside LP have the same height. The third, more important, is that vertical character strokes having similar width are distributed almost identically in LP. The dense vertical edge areas which have some typical perceptual features can be used as a key feature to detect the desired LPs. There can be no doubt that our LBTD descriptor can describe readily the features of LP regions in this sense. Flowchart of the proposed LP localization system is shown in Fig. 5. It consists of three main modules: preprocessing, potential LPs localization and LPs verification. Detailed steps are mentioned in the right part of the schema.

Before going into the descriptive of our system, it is interesting to consider the sub-module motion analysis in case of video sequence input. The goal is to select the best
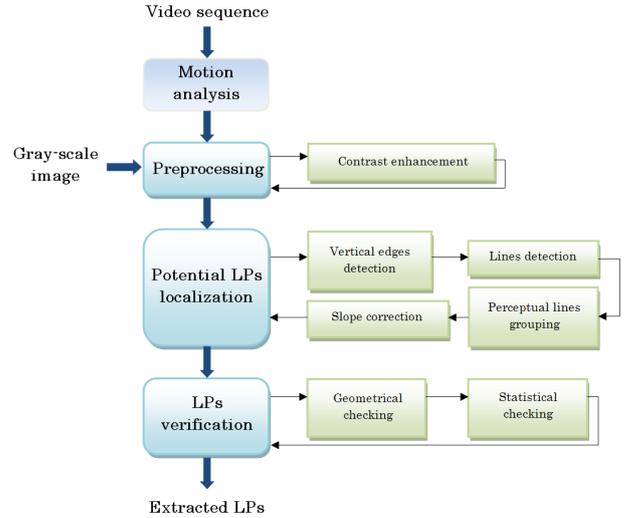

Fig. 5.   Flowchart of proposed LPL system

image in the sense of LP quality. Once moving pixels are detected, an analyzer is activated. Motion is tracking by optical flow technique. When the motion slows down below than a threshold, current frame is selected like a good image for LPL system. So, the blurring effect provoked by moving vehicle could be reduced. It is clear that this idea is suitable to gateway control applications.

A. Preprocessing

To promote our contrast-based method, first, a histogram equalization is carried out to normalize the illumination, using Eq. (x). On the one hand, contrast is increased, on the other hand, vital edges are also protected.

$$F(i) = tanh(\frac{c}{\delta}i) \qquad (7)$$

where i represents the intensity value of the input image, δ is the variance of the image, and c is a constant. An example of this processing step is showed in the Fig. 6.
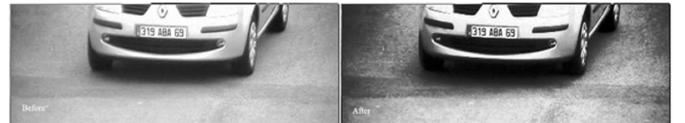

Fig. 6.   An example of contrast enhancement

B. Potential LPs localization

1) Step 1: vertical edges detection: As a fast and simple operator, we used Sobel to obtain the gradient of the input image in vertical direction

2) Step 2: lines detection: The next step is applying the line detection algorithm which is mentioned in the previous section to gray-scale edge image. Each line segment is stored with its extreme coordinates (x_0, y_0), its local orientation θ', and its length l. It is worth recalling here that our algorithm can detect line segments in imperfect image, with accuracy

and rapidity. There is a fundamental difference between our algorithm and other lines detection algorithms, is that no threshold is needed to binarize the edge image.

3) Step 3: perceptual line segments grouping: Firstly, local information of line segments is transformed in global information $[(x_0, y_0), \theta, r, l, r_a]$. Then, the perceptual line segments grouping based on LBTD is performed as in the pseudo-code of Algorithm 1.

Algorithm 1 Perceptual line segments grouping

  for line segments whose angle satisfies $80 \leq \theta \leq 100$ **do**
    repeat
      consider a pair of line segments S_i, S_j
      if $S_i.r_a \sim S_j.r_a$ & $S_i.l \sim S_j.l$ & $|S_i.r - S_j.r| \leq l$
      then
        Two line segments are grouped
      end if
    until all line segments are grouped
    Remove groups with small number of line segments
  end for

---

4) Step 4: slope correction: For each relevant group of segments, a binarized image of segment extremes is generated. Then, a second line detection is applied in this image, with a sweeping range less than 30°. This range is reasonable by the truth that vehicle LPs are positioned almost horizontally. Finally, the angle α in which longest segment was detected is an estimation of LP slope. An inverse rotation of $-\alpha$ degree will correct the inclination. Fig. 7 gives us an example of LP 's slope estimation and rectification.

Fig. 7.   Slop evaluation and correction

As a contribution, this module provides a fine image of characters in LPs to the segmentation part, so improves the recognition performance. Additionally, in the context of fixed camera and moving vehicle, the slope of LP could reflect the direction of the vehicle in a given lane.

C. LPs verification

After the localization process, each one of the candidates must be checked by certain criteria to see if it satisfies the requirements of a LP. Two criteria are adopted sequentially, the geometric properties of LP are utilized first, and then statistical characteristics of the alpha-numerals in the LP are employed in the process of verification.

1) Step 1: geometrical verification: The LPs have uniform size and shape, and the distance between the video camera and the observed vehicle usually falls within a known range. Therefore, the width, height and aspect ratio of the LP image can be estimated and used as a good criterion for selection of true LPs. The width and height threshold are up on applications, while the aspect ratio is always between 2 and 5.

2) Step 2: statistical verification: As mentioned above, vertical strokes in LP have similar height and are distributed almost identically in LP background. Interestingly, a simple check by three horizontal scanning near to top, middle and bottom of each bounding box candidate in the original image can reveal this information. By this way, the number of sudden changes is taken into account for each scanning. If these three values are similar, the candidate is selected as a LP.

To take an example, Fig. 8 shows a process of our LPL system. After preprocessing, the vertical Sobel operator is applied to orignial image for obtaining edge image 8(b). Secondly, line segments are detected, stored and represented in image 8(c). As a result of line segment features extraction and perceptual grouping, image 8(d) shows potential LP candidates behind some white masks. Empirical verification of LPs is involved which give us the locations of two true LPs in this image 8(e). Finally, two LPs are extracted and showed in image 8(f)
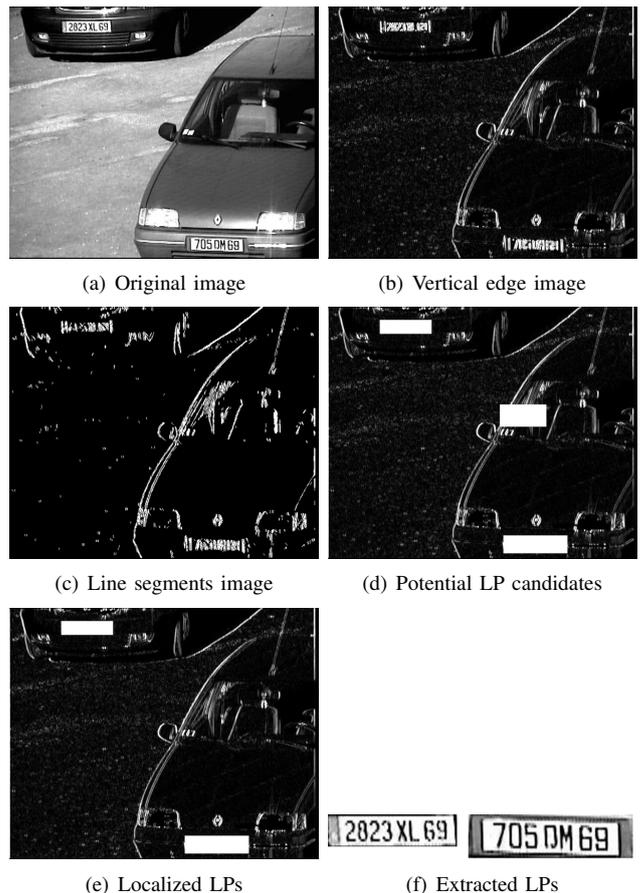

(a) Original image   (b) Vertical edge image

(c) Line segments image   (d) Potential LP candidates

(e) Localized LPs   (f) Extracted LPs

Fig. 8.   An example of step by step results of proposed method

V. EXPERIMENTS AND EVALUATION

In order to analyze the effectiveness of the proposed approach, experiments have been carried out on two image databases. The first one (DB1) consists of 636 gray-scale images taken in sixteen countries for the test set of

Database	RR(%)	PR(%)	FAR(%)	Time(ms)
DB1	97.5	95	5	200
DB2	97	90	10	300

SeeCar project (http://www.htsol.com/Products/Demo.html) This set was built for purpose of testing LP localization and recognition methods in the world-wide applications. All images are of size 768×288 or 768×576. The second one (DB2) was constructed by us-self in the goal of comparing and evaluating LP localization methods. Images are taken from viewpoints in various contexts (parking, gateway, street scene), in different illumination (day, night) and weather (clear, cloudy, rainy) conditions. More specifically, our database contains 600 images whose sizes are between 756×504 and 1024×768.

The ground-truth (GR) of LPs in the two databases is made semi-automatically by using ViPER-GT (http://viper-toolkit.sourceforge.net). ViPER-GT is a tool for annotating videos with metadata, ranging from things like genre and location of cuts to object localization and tracking. The GR information is then stored in corresponding XML file.

In the evaluation stage, recall rate (RR), precision rate (PR), false-alarm rate (FAR) and computation time are used for measuring the overall performance. For this purpose, we impose that a LP is well localized if the overlapping part of detected region and GR occupies more than 80% of GR area.

(a) At night (b) Multi LPs
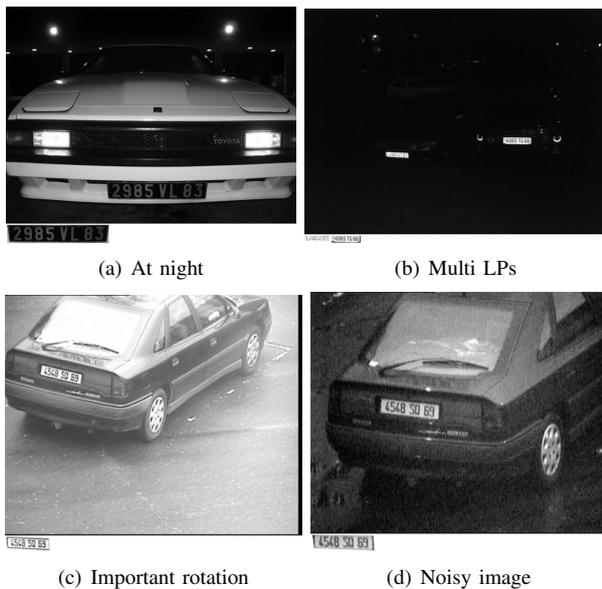

(c) Important rotation (d) Noisy image

Fig. 9.   Robustness of our method in difficult contexts

The proposed system was implemented on a personal computer with Centrino Duo 2GHz, 2GB RAM using C++. The test was conducted on all images in the two databases, and the result of average performance is showed in the Table I. Obtained capacity of localization (97%) is better than

Fig. 10.   An example of missed and wrong localization

the mean of localization performances in the stat of the art (95%) which were tested on databases of similar variability with ours.

The high performance with the DB1 (97.5%) proves that our method can be applied for many countries LP localization applications. Whereas the high performance with the DB2 (97%) means that our method is robust to illumination change, noise, LP orientation and size. Our system performs exact LP localization even in difficult cases: at night, multi LPs with different sizes, LP is inclined about 30°, or noisy image (Fig. 9).

It is interesting to consider that the FAR stills important (10%) in the case of DB2 because of LP-like texture regions, as phone number in the car background . Additionally, partial localization happened when LP have two rows with small number of characters. Fig. 10 shows an example of limitation of our method.

VI. CONCLUSIONS

We presented a real-time and robust License Plate Localization approach for traffic control applications. This approach is based on a new, scale and rotation invariant, texture descriptor which describes the regularity, similarity, directionality and alignment of a line segments. Our approach includes a fast descriptor extraction based on an improvement of connective Hough transform. The tests conducted on real scene images shows that our approach is insensitive to illumination, insensitive to geometric change induced by viewpoint change (scaling, orientation, deformation).

The focus of our future work is the further investigation to decrease false alarms. For this purpose, the empirical verification will be replaced by a learning verification using geometrical features, statistical features and character-based Haar features. On the other hand, we plan to combine extracted LPs with motion estimation for predicting regions of interest (ROIs) and LPs tracking, which make our system more real-time. Finally, more exhaustive comparisons would be conducted.

REFERENCES

[1] M. Ardabilian and L. Chen. A new line segment extraction algorithm: Fast connective hough transform (fcht). In *6th International Conference Pattern Recognition and Information Processing*, 2001.

[2] B. Bouaziz, W. Mahdi, M. Ardabilian, and A. B. Hamadou. A new approach for texture features extraction: Application for text localization in video images. In *IEEE International Conference on Multimedia and Expo*, pages 1737–1740, 2006.

[3] C. C. Chen, J. W. Hsieh, and Y. S. Chen. Detecting license plates from videos using morphological operations and adaboosting algorithm. In *International Computer Symposium*, volume 3, pages 1201–1204, 2006.

[4] H. Chen, J. Ren, H. Tan, and J. Wang. A novel method for license plate localization. *4th International Conference on Image and Graphics*, pages 604–609, 2007.

[5] H. Chen, D. Rivait, and Q. Gao. Real-time license plate identification by perceptual shape grouping and tracking. *The 9th International IEEE Conference on Intelligent Transportation Systems*, pages 1352–1357, 2006.

[6] G. Heo, M. Kim, I. Jung, D.-R. Lee, and I.-S. Oh. Extraction of car license plate regions using line grouping and edge density methods. *International Symposium on Information Technology Convergence*, 0:37–42, 2007.

[7] B. Hongliang and L. Changping. A hybrid license plate extraction method based on edge statistics and morphology. In *17th International Conference on Pattern Recognition*, volume 2, pages 831–834, 2004.

[8] C.-T. Hsieh, Y.-S. Juan, and K.-M. Hung. Multiple license plate detection for complex background. In *19th International Conference on Advanced Information Networking and Applications*, volume 2, pages 389– 392, 2005.

[9] W. Jia, H. Zhang, and X. He. Region-based license plate detection. *Journal of Network and Computer Applications*, 30(4):1324–1333, 2007.

[10] V. N. Koval, V. Turchenko, V. Kochan, A. Sachenko, and G. Markowsky. Smart license plate recognition system based on image processing using neural network. In *Second IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, pages 123–127, 2003.

[11] H. Mahini, S. Kasaei, and F. Dorri. An efficient features - based license plate localization method. In *The 18th International Conference on Pattern Recognition*, volume 2, pages 841–844, 2006.

[12] J. Matas and K. Zimmermann. Unconstrained licence plate and text localization and recognition. In *IEEE Proceeding on Intelligent Transportation Systems*, pages 225 – 230, 2005.

[13] W. Peng, R. Yong Man, W. Chee Sun, and C. Yanglim. Texture descriptors in mpeg-7. In *Proceedings of the 9th International Conference on Computer Analysis of Images and Patterns*. Springer-Verlag, 2001. 753325 21-28.

[14] F. Porikli and T. Kocak. Robust license plate detection using covariance descriptor in a neural network framework. In *International Conference on Video and Signal Based Surveillance*, 2006.

[15] A. Pujol and L. Chen. Hough transform based cityscape classifier. In *6th International Workshop on Image Analysis for Multimedia Interactive Services*, 2005.

[16] V. Sharpiro and D. Dimov. Adaptive license plate image extraction. In *5th International Conference on Computer Systems and Technologies*, 2003.

[17] D. T. Tran, L. H. D. Tran, P. V. Tran, and V. H. Nguyen. Building an automatic vehicle license-plate recognition system. In *3rd International Conference in Computer Science: Recherche, Innovation et Vision du Futur*, 2005.

[18] P. Wu, H.-H. Chen, R.-J. Wu, and D.-F. Shen. License plate extraction in low resolution video. In *18th International Conference on Pattern Recognition*, 2006.

[19] S. M. Youssef and S. B. AbdelRahman. A smart access control using an efficient license plate location and recognition approach. *Expert Systems with Applications*, 34(1):256–265, 2008.

[20] H. Zhang, W. Jia, X. He, and Q. Wu. Real-time license plate detection under various conditions. In S. B. . Heidelberg, editor, *Ubiquitous Intelligence and Computing*, volume 4159, pages 192–199. 2006.