

A Multi-style License Plate Recognition System based on Tree of Shapes for Character Segmentation

Francisco Gómez Fernández¹, Pablo Negri², Marta Mejail¹, and Julio Jacobo¹

¹ Departamento de Computación-Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires, Argentina fgomez [a] dc.uba.ar
<http://www-2.dc.uba.ar/grupinv/imagenes/>

² PLADEMA-Universidad del Centro de la Provincia de Buenos Aires
Tandil, Argentina pnegri [a] exa.unicen.edu.ar
<http://www.pladema.net>

Abstract. The aim of this work is to develop a multi-style license plate recognition (LPR) system. Most of the LPR systems are country-dependent and take advantage of it. Here, a new character extraction algorithm is proposed, based on the tree of shapes of the image. This method is well adapted to work with different styles of license plates, does not require skew or rotation correction and is parameterless. Also, it has invariance under changes in scale, contrast, or affine changes in illumination. We tested our LPR system on two different datasets and achieved high performance rates: above 90 % in license plate detection and character recognition steps, and up to 98.17 % in the character segmentation step.

1 Introduction

License Plate Recognition (LPR) is a very popular research area because of its immediate applications in real life. Security control and traffic safety applications, such identification of stolen cars or speed limit enforcement, have become very important application areas where the license plate (LP) analysis plays a fundamental role [1].

An LPR system can be divided in three steps: LP detection, character segmentation and character recognition. Character recognition success strongly depends on the quality of the bounding boxes, obtained by the segmentation step. Therefore, we considered that segmentation is a very important step in an LPR system. An extensive review for LPR can be found in [1]. However, the problem of LPR systems able to handle license plates from different countries and with different styles (shape, foreground-background colors, etc.) is currently an open research area. Several works implement LPR tasks achieving high performance rates, but most of them are country dependent.

In [6,10,11] LPR with multi-style analyses is addressed. Also, [6] and [11] use a similar procedure to search for LP regions, and added a recognition feedback to improve the detection step when the recognition fails. The character

extraction step, is usually performed by binarization methods and a connected component analysis [10,11]. The choice of binarization-thresholds is a hard task; if it is not chosen properly, we will easily get redundant detections or miss some detections too. An interesting work which handled detection and segmentation simultaneously is presented in [4].

In [11], the recognition step is carried out by a statistical approach using Fourier descriptors, and a structural approach using the Reeb Graph to distinguish ambiguous characters. In addition, for better character recognition, in [6] a three-layer artificial neural network over fixed sub-blocks from previously extracted characters, is computed.

In this work we develop a LPR System on still images adaptable to different countries. Our focus is in the segmentation step which is considered to be very important in an LPR system. A new character extraction method is proposed based on the tree of shapes of an image. This method is well adapted to work under different LP styles, does not require rotation or skew correction and is parameterless. Also, it has invariance under changes in scale, contrast, or affine changes in illumination. These properties are derived by the properties of the tree of shapes [8]. The system was tested on two datasets (see examples in Fig. 1) obtaining high performance rates.

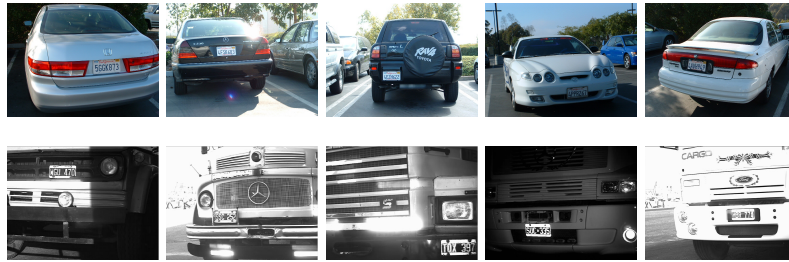


Fig. 1: Examples from two datasets used to test our system. First row shows cars images from USA. Second row shows Argentinean truck images.

This paper is organized as follows. Section 2 details the implementation of the LPR system and its steps. Experimental results over the datasets are given in section 3. Finally, section 4 presents the conclusions and future work.

2 License plate recognition system

In this section we introduce the three steps of the LPR system: license plate detection, character segmentation and character recognition (Fig. 2).

The initial task of any LPR system is to find the location of the LP in the image. Thus, our LP detection process starts generating several regions of interest (RoI) using morphological filters. To validate the RoIs $R_i, i = 1, \dots, N$ and choose the most probable LP region, more exhaustive analyses are applied to

give a score to each region using template matching and feature extraction [9]. Then, the system passes the region with the highest score to the segmentation step and validates its result if it has encountered more than three bounding boxes. Finally, the bounding boxes are used as an input to the character recognition step and it is validated as described in section 2.3. Following [11], if the analysis fails in the character segmentation or character recognition steps, the second most probable region will be evaluated, and so on, until the R_N region is reached. In such situation, the system returns no detection.

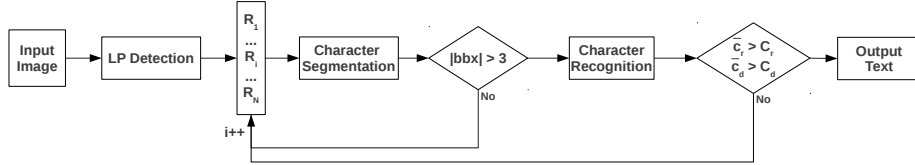


Fig. 2: LPR system diagram. Diamond shaped blocks represent validation steps.

2.1 License plate detection

In this section we discuss the analysis done for every region in order to give them a confidence value.

ROI generation A morphological top-hat filtering is applied to the input image to enhance the contrast in regions with great difference in intensity values. Then, vertical contours are calculated using the Sobel filter, and successive morphological operations are then applied to connect the edges in the potential LP regions. These operations are a simple and rapid way to provide several potential RoIs. This is a critical step in the LPR system: if the LP is not detected by the morphological filters, it will be lost.

ROI evaluation Each RoI R_i , $i = 1, \dots, N$ is evaluated by two methods: template matching and text detection [5]. Then, we define four evaluation vectors of length N : \mathbf{pcv} for template matching, and \mathbf{mgd} , \mathbf{nts} and \mathbf{tbr} for text detection, where $\mathbf{pcv}(i)$ is the pattern correlation value obtained by cumulating the correlation values inside the boundaries of R_i , and $\mathbf{mgd}(i)$, $\mathbf{nts}(i)$, $\mathbf{tbr}(i)$ are the maximum gradient difference, the number of text segments and the text block ratio, inside R_i . We need to merge their information in order to decide which of the N regions is the most probable to be a license plate. To do so, we create four sorting index vectors: \mathbf{pcv}^{si} , \mathbf{mgd}^{si} , \mathbf{nts}^{si} and \mathbf{tbr}^{si} . These vectors give an index to each R_i that depends on an ascending sorting of the evaluation vectors. The R_i with the lowest value in the feature vector gets index 1, and the R_i with the highest value gets index N . Then, we define a vector \mathbf{votes} of length N :

$$\mathbf{votes}(i) = \mathbf{pcv}^{\text{si}}(i) + \mathbf{mgd}^{\text{si}}(i) + \mathbf{nts}^{\text{si}}(i) + \mathbf{tbr}^{\text{si}}(i)$$

The region R_m , with $m = \arg \max_{1 \leq i \leq N} \mathbf{votes}(i)$ is retained as the LP.

Adapting the thresholds The detection step can be applied to different datasets. However, the thresholds must be adapted to detect the LP in images obtained from different cameras or environments. To do this, the first images of the dataset are analyzed. As the position of each license plate is labeled in each image of the dataset, this information is used in order to fix the thresholds of the detection method. These thresholds are chosen to validate 95 % of all text segments inside the first five well recognized LPs. Also, the LP used as correlation pattern is set to the first image of the dataset.

2.2 Character Segmentation

To extract the characters in the LP we propose a new algorithm, which processes the tree of shapes of an image [8] to search for groups of characters. The tree of shapes is a complete representation of an image, i.e. the original image can be reconstructed from it. Also, the shapes in the tree are consistent with what we expect to be “objects” in the image. For instance, a character in an image will be represented by a shape (or a set of shapes) in the tree. The goal of this procedure is to state properties shared by every LP with no restrictions on the style of the plate.

Tree of shapes A shape is defined as a connected component of a level set whose holes are “filled” (see [8] for a formal definition). Then, upper and lower level sets, at level λ , of an image u are defined as $\mathcal{X}_\lambda u = \{x \mid u(x) \geq \lambda\}$ and $\mathcal{X}^\lambda u = \{x \mid u(x) < \lambda\}$, respectively. It is known that connected components of level sets can be arranged in an inclusion tree ordered by λ , their gray level [8]. Moreover, the shapes extracted from an image can be ordered by geometrical inclusion (a shape is a child of another shape if it is included in its interior) to build the tree of shapes.

Char-grouping algorithm This algorithm uses the fact that characters in license plates have properties in common, such as same foreground-background contrast, alignment and minimum overlap of bounding boxes, and similar width and height. The steps of this algorithm can be summarized as follows. The RoI R_m , returned by the detection step, is used to compute the tree of shapes. Then, all the nodes (shapes) in the tree are pairwise compared, linking the similar shapes, with a given criterion. Finally, the bounding boxes of the most linked node and its neighbors are returned as the result. Algorithm 1 shows the pseudo-code of the proposed algorithm.

In order to avoid performing comparison of shapes included in other shapes or in already linked shapes, the tree of shapes is traversed taking advantage of its structure: nodes are visited top-down, and a node is never compared with a descendant of it or with a descendant of a node linked to it (line 4). This is a proper traverse, i.e. there are no repeated nor missing comparisons, due to the

inclusion property of the tree of shapes, and to the fact that any two shapes are either disjoint or nested (see [8] for an explanation).

Character comparison A feature vector is built for each node in the tree. This feature vector bears information about the bounding box and the type (upper or lower) of its associated shape. The comparison of the feature vectors (line 2) is carried out by the predicate `SIMILARCHARS(n, m)`, which returns *true* if nodes n and m have the same type, and the distance between the corresponding shapes Φ is above a fixed threshold, and returns *false* otherwise. The distance Φ is given by

$$\Phi(n, m) = \frac{\min(\mathcal{W}(n), \mathcal{W}(m))}{\max(\mathcal{W}(n), \mathcal{W}(m))} + \frac{\min(\mathcal{H}(n), \mathcal{H}(m))}{\max(\mathcal{H}(n), \mathcal{H}(m))} + \frac{y(n) \cap y(m)}{\min(\mathcal{H}(n), \mathcal{H}(m))} + 1 - \frac{x(n) \cap x(m)}{\min(\mathcal{W}(n), \mathcal{W}(m))}$$

where functions $\mathcal{W}(\cdot)$ and $\mathcal{H}(\cdot)$ return width and height of the corresponding bounding boxes of n and m , respectively. Also, terms $x(n) \cap x(m)$ and $y(n) \cap y(m)$ represent the bounding box overlapping in x and y directions, respectively. In addition, shapes which lack vertical rectangularity, or which are too small, or too big or too distant, are discarded before performing the comparison.

Algorithm 1 char-grouping algorithm

Input: Tree of shapes \mathcal{T}

Output: Set of bounding boxes \mathcal{S}

```

1: for all  $n, m \in \mathcal{T}$  do
2:   if SIMILARCHARS( $n, m$ ) then
3:     LINK( $n, m$ )
4:     Skip  $n$ 's children and  $m$ 's children
5: Let  $n_{max}$  the maximum linked node
6: for all  $n \in \mathcal{T}$  do
7:   if LINKED( $n, n_{max}$ ) then
8:      $\mathcal{S} \leftarrow \mathcal{S} \cup \{ \text{BOUNDINGBOX}(n) \}$ 
9: return  $\mathcal{S}$ 

```

Fig. 3 shows examples of the bounding boxes computed by the *char-grouping algorithm*. As it can be seen, characters of LPs of very different styles are detected without modifying the algorithm. The first column shows examples of cluttered images where the detected text region is highly over-sized but the segmentation step still succeeds. The second column shows segmentation examples with different foreground-background color combination (top and middle: dark foreground and bright background, bottom: bright foreground and dark background). The third column shows how the contrast invariance property of the tree of shapes gives the advantage to work under several and nonuniform illumination conditions of image acquisition. The fourth column shows segmentations for LPs not in the tested datasets.

As we can see, the *char-grouping algorithm* has no need of rotation or skew correction, it is style independent and furthermore it is parameterless. Also,

it works under changes in the scale of the license plate and under changes in contrast or illumination conditions. These properties are achieved without constraints on the style of license plate or a priori information.



Fig. 3: Examples of character segmentation using *char-grouping algorithm*.

2.3 Character recognition

A Support Vector Machine (SVM) based classifier is trained using the Histogram of Gradient (HoG) as features.

Histogram of Gradient This feature uses gradient magnitude and orientation around a point of interest or in a region of the image to construct a histogram. The HoG feature space is composed of histograms obtained from rectangular regions shifted by one pixel inside the pattern, defined as follows.

Once a character is segmented, it is resized to a 16x12 pixels pattern and then applied a 3x3 Sobel filter. The gradient orientation of each pixel is quantized to integer values between 0 and 5 using modulo π instead of modulo 2π . In this way, dark on bright characters give the same orientation than bright on dark characters. For each pixel p in the pattern, we considered nine regions with p as top-left corner and sizes $M \times M$, $M \times 2M$, $2M \times M$, $M \in \{4, 6, 8\}$ to build the histograms. Then, the histograms are normalized to obtain their sum equals to 1.

Support vector machine In this work, we train a SVM using *libsvm* library [2]. The strategy for the classification is the *One Against All* approach. The training characters are extracted from images labeled as non-deteriorated (see section 3) which are not included in the test dataset. We construct 35 binary (O and 0 are in the same class) SVM classifiers, each of which separates one character from the rest. To get the k -th SVM classifier the training dataset is composed as follows: the positive set correspond to samples from k -th class, and the negative set correspond to samples of other classes. In the testing phase, an input character (resized and normalized) is the input of each classifier. The, it will be classified

as the class that classifier produces the highest value.

Character recognition validation The character recognition results are tested following the strategy developed in [11]. Two confidence values are estimated from the SVM classifiers outputs: c_r and c_d , that indicate classifier performance and discriminability, respectively. For each character, these values are calculated and the total performance is obtained performing the mean of these results, obtaining \bar{c}_r and \bar{c}_d . If $\bar{c}_r < C_r$ and $\bar{c}_d < C_d$, all the operation is invalidated and the region is rejected. The thresholds C_r and C_d are estimated to validate the 99 % of the training dataset.

3 Experimental results

We tested our LPR system performance on two datasets (see examples in Fig. 1). The first dataset, from now on called the USA dataset, is composed of 158 images tagged as non-deteriorated from the UCSD/Calit2 database [3]. These images were captured from outdoor parked cars and the license plates have different styles, containing alphanumeric characters without an established configuration. The second dataset, from now on called the ARG dataset³, is composed of 439 truck images from Argentina. These images were acquired by an infrared camera placed at a truck entrance gate. All the images have the same style, but this style is not used to tune up the system. Both datasets were manually tagged with plate text, plate location and character bounding box. An extra label is added indicating if the image is deteriorated or non-deteriorated, where deteriorated images are those with a license plate which is too noisy, broken or incomplete. To validate a detection we check if the detected region intersects more than the half of the tagged region. In an analogous way, we validate the character segmentation. Also, character recognition is evaluated using the Levenshtein distance.

Additionally, we tested our system using Maximally Stable Extremal Regions (MSER) [7] in the segmentation step. The MSER has been widely used in many applications, including license plate recognition [4]. Two variants of MSER can be computed denoted as MSER+ and MSER-. The first detects bright regions with darker boundary, and the second detects dark regions with brighter boundary. For the purpose of character extraction, we set the sensitivity parameter $\Delta = 10$ and we filter out unstable or repeated regions.

Detection, segmentation and recognition rates for ARG and USA datasets are show in Table 1. The LPR system with *char-grouping algorithm* (CGA) and MSER, achieved similar detection performance rates as expected because the detection step is the same. However, the CGA outperforms MSER in 3 % in the segmentation step and therefore in recognition too, because character recognition strongly depends on the quality of the bounding boxes obtained by the segmentation step. Moreover, the MSER procedure needs information about

³ Available at ...

the foreground-background contrast, e.g. MSER+ for ARG dataset and MSER- for USA dataset, resulting in a loss of the multi-style characteristic of the system. Also, CGA and MSER have better performance on the ARG dataset than on the USA database because of the difference in image acquisition conditions between both datasets.

	Det	Seg	Rec		Det	Seg	Rec
CGA	97.27	98.17	95.08	CGA	90.51	93.76	92.45
MSER+	97.04	95.54	92.18	MSER-	89.12	90.47	89.27

(a) ARG dataset

(b) USA dataset

Table 1: Detection, segmentation and recognition rates, using *char-grouping algorithm* (CGA) and Maximally Extremal Stable Regions (MSER+ and MSER-).

4 Conclusions and future work

This work introduced a novel LPR system for multi-style license plates, which proposed a new algorithm for character extraction. This algorithm does not require rotation or skew correction and is parameterless. Also, it has invariance under changes in scale, contrast, or affine changes in illumination. The quantitative and qualitative results shown in the previous sections, support the mentioned properties. Further work has to be done to study the adaptation of the detection thresholds without any a priori information. Also, we think that adding features to the nodes of the tree of shapes, like pixel distribution inside a bounding box, will enhance the comparison. Moreover, the need to extend the system to handle two-row LP is an important task to tackle in further studies.

References

1. Anagnostopoulos, C., Anagnostopoulos, I., Psoroulas, I., Loumos, V., Kayafas, E.: License plate recognition from still images and video sequences: A survey. *IEEE Intelligent Transportation Systems* 9(3), 377–391 (2008)
2. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines (2001), software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
3. Dlagnekov, L., Belongie, S.: Ucsd/calit2 car license plate, make and model database (2005), http://vision.ucsd.edu/car_data.html
4. Donoser, M., Arth, C., Bischof, H.: Detecting, tracking and recognizing license plates. In: *Asian Conference on Computer Vision*. pp. 447–456 (2007)
5. E. K. Wong, M.C.: A new robust algorithm for video text extraction. *Pattern Recognition* 36, 1397–1406 (2003)
6. Jiao, J., Ye, Q., Huang, Q.: A configurable method for multi-style license plate recognition. *Pattern Recognition* 42(3), 358 – 369 (2009)

7. Matas, J., Chum, O., Martin, U., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: BMVC. vol. 1, pp. 384–393 (2002)
8. Monasse, P.: Morphological representation of digital images and application to registration. Ph.D. thesis, Université Paris IX-Dauphine (June 2000)
9. Negri, P., Tepper, M., Acevedo, D., Jacobo, J., Mejail, M.: Multiple clues for license plate detection and recognition 6419, 269–276 (2010)
10. Shapiro, V., Gluhchev, G., Dimov, D.: Towards a multinational car license plate recognition system. MVA 17, 173–183 (2006)
11. Thome, N., Vacavant, A., Robinault, L., Miguet, S.: A cognitive and video-based approach for multinational license plate recognition. MVA 21 (2010)