A New Block-Wise Algorithm for License Plate Location

M. Saadatmand-Tarzjan*, V. Nikzade**, H. Ghassemian***
*Tarbiat Modares University, Tehran, Iran. saadatmand@kiaee.org
**Binatooth Co., Khorasan Science and Technology Park, Mashhad, Iran. nikzade@gmail.com
***Tarbiat Modares University, Tehran, Iran. ghassemi@modares.ac.ir

Abstract: In this paper, a new block-wise algorithm is proposed for Iranian license plates location. The proposed algorithm uses the density of vertical edges (DVE) in the license plate region as a feature for detecting the candidate license plates. It employs the quad-tree decomposition algorithm to adjust the size of the blocks used for computing local DVEs. According to the experimental results, the algorithm is fairly robust against variations of imaging geometry (e.g., scale and orientation) and illumination conditions. Furthermore, the proposed algorithm provided better performance compared to a well-known method while it was significantly more efficient.

Keywords: License Plate Location, Block-Wise Algorithms, Quad-tree Decomposition, Edge Detection.

1. Introduction

License plate recognition (LPR) is an important technique in intelligent transportation systems (ITS) such as automatic payment of tolls on highways [1], parking lots [2], travel-time data provision [3], and traffic law enforcement [4]. Each LPR system consists of three tasks [5]: vehicle detection, plate locating, and license number identification (recognition). The first task detects vehicles in a sequence of images [6]. The plate location algorithms determine candidate license plate regions in the image [7]. Finally, the license number recognition task identifies the license plate number of each vehicle in the image [8].

Among the above tasks, the plate location may be the most important one which has been received considerable attention [7, 9]. Plate detection is the most computationally intensive task because whole image should usually be processed in order to detect the plate. Furthermore, it should be robust against different environmental conditions such as various backgrounds especially in outdoor scenes (non-stationary backgrounds) [5], changed illumination [7], and wide ranges of the distance between camera and vehicle (wide distance ranges) [10].

Up-to-date, using different features of license plates, a large number of methods have been proposed for plate location. Features commonly employed have been derived from the license plate format and alphanumeric characters constituting license numbers [11]. The features regarding the license plate format include the shape [12], symmetry [13], width to height ratio [14], abundance of vertical edges in the license plate region [7], plate boundaries [15], color [11], gray texture [9], spatial frequency [16] and variance of intensity values [17]. Character features include the local line-type shape and global blob-type shape of characters [18], high contrast of characters against the license plate background [19], aspect ratio of characters [20], and alignment of characters [5].

Generally, the plate location algorithms can be divided to three different categories: pixel-wise, character-wise and block-wise methods. The pixel-wise methods use features obtained by low-level processing of pixels [12, 14]. In a license plate, characters are aligned in the same direction. This fact makes the main idea of character-wise methods [5, 11, 18].

In this paper, we focus on the block-wise methods in which the candidate plate areas are distinguished through the features extracted from different blocks of the image [6, 7, 9, 10, 21]. For example, Kim et al [6] developed a learning based approach in which two time-delay neural networks (TDNNs) are used as filters to analyze the color and texture properties of the license plate. Therefore, its performance somehow depends on the scene illumination. In a similar work, Park et al [21] used neural networks as filters for processing small windows. Zunino and Rovetta [9] proposed an algorithm based on vector quantization (VQ) in which a quadtree representation is used by a specific coding mechanism. This representation can provide some information about the contents of image regions which can boost location performance. Recently, Zheng et al [7] devised an algorithm based on the local density of vertical edges (DVE). In this algorithm, the vertical edges are first extracted from the enhanced image using Sobel operator. After removing the non-plate edges, entire edge image is filtered by a window whose elements are set to one to count the total number of edge points in the window. If it is above a certain percentage of the window area, there will likely be a license plate. Finally, Yamaguchi et al [10] proposed an algorithm based on template matching for route bus identification.

Most of the previous works, in some way, restricted the working conditions. In more details, they usually use
fixed-size blocks to detect the license plate [6, 7, 10, 21]. It means that their performance may be degraded by changing the scale. Furthermore, the plate size is usually used in order to determine the real license plate region from the other candidate areas [7, 9-11]. This parameter is exactly dependent on the scale and imaging geometry. In other word, these algorithms may disappointingly perform for incline license plates. Thus, developing a plate location algorithm for outdoor scenes with non-stationary backgrounds, wide distance ranges, different view-points and changed illumination remains a challenging endeavor.

In this paper, we propose a new block-wise algorithm for Iranian license plates location. It uses high DVE as a feature for detecting candidate license plate areas. The main advantage of the proposed algorithm is employing blocks with dynamic size. In the first step, the proposed algorithm effectively extracts vertical edges of the image. Then, it takes advantage of the quadtree decomposition algorithm in order to divide the image into small blocks based on the local DVE. Subsequently, a density map is obtained by computing normalized number of vertical edges points in each block. After segmentation of the density map by means of an appropriate threshold, the candidate plate areas are obtained by removing non-plate components. Finally, the best candidate region is chosen as the license plate region in the image.

This paper is organized as follows. The general types of Iranian license plates are introduced in Section 2. Section 3 explains the proposed algorithm for license plate location. Experimental results are given in Section 4 and finally, Section 5 is devoted to concluding remarks.

3. Proposed Algorithm for Plate Location

The best feature for Iranian license plate location is DVE since it is common for all Iranian plate types with any background color. Therefore, a scale-invariant block-wise method should be developed to detect the regions with high DVE in the image. As shown in Fig. 2, the proposed algorithm includes four steps: edge detection, density map generation, thresholding and the best region selection.

3.1 Vertical Edge Detection

For vertical edge detection, the input gray-level image is first smoothed using median filter [22] in order to suppress noise. A 3×3 vertical Sobel filter [22] is applied to the resultant image (Fig. 3.b). The redundant edge coefficients may abnormally increase DVE. To overcome this problem, we suppress non-maximum edge coefficients in the image along the horizontal direction [23] (Fig. 3.c). Finally, the resultant image is binarized by a global threshold obtained by Otsu’s method [24] (Fig. 3.d).

3.2 DVE Map Generation

The license plate is a region with high DVE in the image. We can compute the local density by simply counting the number of edge pixels in a moving window on the image. However, a fixed-size window may degrade the algorithm performance for incline plates and make it sensitive to the scale. In order to overpower this problem, we use the quadtree decomposition algorithm [11]. In this approach, the whole image is initially considered as one block. Then, in each block, if DVE is more than a specific threshold (δDVE), it will be divided into four equal-size sub-blocks. The above process is continued until the above criterion is satisfied or the block size becomes less than a specified threshold (δsize). In more details, we regulate the moving window based on the local DVE. Therefore, in the areas with high DVE, the size of the blocks will be smaller compared to the
smooth regions. The DVE map is obtained by computing the average number of vertical edge pixels in each block as illustrated in Fig. 3.e.

3.3 Thresholding

To extract the candidate plate area, the DVE map should be binarized. We use a global threshold obtained by Otsu’s method for this aim as shown in Fig. 3.f.

3.4 Best Region Selection

In order to determine the plate region, first, unacceptable candidate areas are ignored based on their size, aspect ratio, and distance from the image margins. Then, each candidate region is binarized using a threshold obtained by Otsu’s method in order to extract characters. After removing unacceptable characters based on their size and aspect ratio, the proportional area of the remaining characters with respect to the total region area is computed.

Finally, the candidate regions are sorted based on the proportional area and the best region (with the maximum proportional area) is chosen as the license plate area as illustrated in Fig. 3.g.

Experimental Results

We used a reference image database (imagebase) including 651 images to study the performance of the proposed algorithm. This imagebase contains all general categories of Iranian license plates as shown in Table I.

Fig. 4 illustrates the results of the proposed algorithm for an Iran license plate, imaging from four different view-points. Despite significant variations of the scale and orientation (caused by changing imaging geometry), the algorithm successfully extracted the license plate in all illustrations. As shown in Figs. 5 and 6, our algorithm could successfully locate the lasery license plates in the images with different scale, orientation, and illumination conditions. Similar results are also drawn for old license plates in Fig. 7.

As illustrated in Fig. 8, the proposed algorithm is appropriate for license plates location in outdoor images, because it is fairly robust against changes of illumination conditions and imaging geometry.

We compared the performance of our algorithm with that of Zheng’s method [7]. For each image, we indicated two plate candidates, and the percentage of license plates hit by the first and second candidates are listed in Table II. As shown, both of the algorithms provide good percent rates. In more details, the proposed algorithm hits a larger percentage of license plates by the first candidates. The difference between the percent rates of both algorithms for non-detected license plates is negligible as 0.2%. Furthermore, the proposed algorithm was significantly (about 40 times) faster than Zheng’s method. Therefore, the new algorithm is more efficient and provides better performance.

<table>
<thead>
<tr>
<th>License plate type</th>
<th>Number of images</th>
</tr>
</thead>
<tbody>
<tr>
<td>Old type</td>
<td>95</td>
</tr>
<tr>
<td>Lasery type</td>
<td>489</td>
</tr>
<tr>
<td>Iran type</td>
<td>68</td>
</tr>
<tr>
<td>Total</td>
<td>651</td>
</tr>
</tbody>
</table>

Table I. Supported license plate types in the reference imagebase.
method to adjust the size of density computation blocks. Experimental results demonstrated that the algorithm is suitable for outdoor images since it is fairly robust against variations of illumination, scale and orientation. Furthermore, it gave better performance compared to Zheng’s method.

Acknowledgements
The authors thank NRP Co. for initiating and supporting this work.

References
Fig. 8. Results of the proposed algorithm for four outdoor images.


Table II. Comparison of location rates of the proposed algorithm and Zheng’s method. The best results have been shown by bold-faced text.

<table>
<thead>
<tr>
<th>Methods</th>
<th>First Candidates</th>
<th>Second Candidates</th>
<th>Plates not detected</th>
<th>CPU Time (Sec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed algorithm</td>
<td>98.9%</td>
<td>0.3%</td>
<td>0.8%</td>
<td>1.2</td>
</tr>
<tr>
<td>Zheng’s method</td>
<td>96.3%</td>
<td>3.1%</td>
<td>0.6%</td>
<td>47.9</td>
</tr>
</tbody>
</table>