Available online at www.sciencedirect.com

SCIENCE DIRECT®

Pattern Recognition Letters 26 (2005) 2431–2438

Pattern Recognition Letters

www.elsevier.com/locate/patrec

An efficient method of license plate location

Danian Zheng *, Yannan Zhao, Jiaxin Wang

State Key Laboratory of Intelligent Technology and Systems, Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China

Received 11 May 2004; received in revised form 4 April 2005
Available online 22 June 2005

Communicated by H. Wechsler

Abstract

License plate location is an important stage in vehicle license plate recognition for automated transport system. This paper presents a real time and robust method of license plate location. License plate area contains rich edge and texture information. We first extract out the vertical edges of the car image using image enhancement and Sobel operator, then remove most of the background and noise edges by an effective algorithm, and finally search the plate region by a rectangle window in the residual edge image and segment the plate out from the original car image. Experimental results demonstrate the great robustness and efficiency of our method.

Keywords: Image enhancement; Edge detection; Long curve and random noise removing; Plate location and segmentation

1. Introduction

Nowadays license plate recognition becomes a key technique to many automated transport systems such as road traffic monitoring, automatic payment of tolls on highways or bridges and parking lots access control. License plate location is an essential and important stage in this technique, and it has received considerable attention.

Researchers have found many diverse methods of license plate location. Rodolfo and Stefano (2000) devised a method based on vector quantization (VQ). VQ image representation is a quadtree representation by the specific coding mechanism, and it can give a system some hints about the contents of image regions, and such information boosts location performance. Park et al. (1999) used neural networks to locate license plate. Neural networks can be used as filters for analyzing small windows of an image and deciding whether each window contains a license plate, and their inputs are HSI values; a post-processor combines

* Corresponding author. Tel.: +86 10 62775613; fax: +86 10 62795871.
E-mail address: zdn02@mails.tsinghua.edu.cn (D. Zheng).

0167-8655/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2005.04.014

these filtered images and locates the bounding boxes of license plates in the image. Besides neural networks, other filters have been considered too. For example, some authors used line sensitive filters to extract the plate areas. License plates are identified as image areas with high density of rather thin dark lines or curves. Therefore, localization is handled looking for rectangular regions in the image containing maxima of response to these line filters, which is computed by a cumulative function (Luis et al., 1999). Plate characters can be direct identified by scanning through the input image and looking for portions of the image that were not linked to other parts of the image. If a number of characters are found to be in a straight line, they may make up a license plate (Lim et al., 1998). Fuzzy logic has been applied to the problem of locating license plate by Zimic et al. (1997). The authors made some intuitive rules to describe the license plate, and gave some membership functions for the fuzzy sets "bright" and "dark", "bright and dark sequence" to get the horizontal and vertical plate positions. But this method is sensitive to the license plate color and brightness and needs much processing time. Using color features to locate license plate has been studied by Zhu et al. (2002) and Wei et al. (2001), but these methods are not robust enough to the different environments. Edge features of the car image are very important, and edge density can be used to successfully detect a number plate location due to the characteristics of the number plate. Ming et al. (1996) developed a method to improve the edge image by eliminating the highest and lowest portions of the edge density to simplify the whole image. But some of the plate region identity will be lost in this method.

This paper further researches the subject of license plate location. The rectangle license plate contains rich edge and texture information, so we consider it in its edge image but very different to Ming et al. (1996). We first enhance the original car image to boost up the plate area, then extract the vertical edge image using Sobel operator, and then remove the background curves and noise in the edge image, and finally slide a rectangle window to search the plate in the residual image and segment it out from the original car image. Section 2 describes our method of license plate location, and it contains four parts: image enhancement, vertical edge extraction, background curve and noise removing, plate search and segmentation. Experiments with three sets of car images are performed in Section 3. Section 4 gives the discussion and conclusions.

2. The proposed method for license plate location

All the input car images have 384×288 pixels and 256 gray levels, and an example image is given in Fig. 1. The license plate of the car consists of several characters (such as Latin letters, Arabic numerals, etc.), so the plate area contains rich edge information. But sometimes the background of the car image holds much edge information too. There are two facts that attract our attention: one is that the background areas around the license plate mainly include some horizontal edges; the other is that the edges in the background are mainly long curves and random noises, whereas the edges in the plate area cluster together and produce intense texture feature. If only the vertical edges are extracted from the car image (although the plate will lose a little horizontal edge information, this little loss is to be valuable) and most of the background edges are removed, the plate area will be isolated

Fig. 1. An example car image with 384×288 pixels and 256 gray levels.

out distinctly in the whole edge image. Thus we propose to locate the license plate in its vertical edge image as the following four stages.

2.1. Image enhancement

In Fig. 1, the gradients in the license plate area are much lower than those in the contour areas of the car, which is caused by the car shadow in the dazzling sunshine. The car images captured in the gloomy days or dim nights often bring out weak gradients in plate areas too. A few vertical edges will appear in the plate areas, if we extract edge images directly from these car images. Therefore it is important to enhance the car images firstly.

The local areas that need to be enhanced in a car image have low variances. Here we use $I_{i,j}$ to denote the luminance of the pixel $P_{i,j}$ (row: $0 \leqslant i < 288$, column: $0 \leqslant j < 384$) in the car image, and use $I'_{i,j}$ to denote the luminance in the enhanced image. We let $I_{i,j}$ and $I'_{i,j}$ satisfy Eq. (1), where $W_{i,j}$ is a window centered on pixel $P_{i,j}$, $\overline{I}_{W_{i,j}}$ and $\sigma_{W_{i,j}}$ are the mean luminance and standard deviation of the pixels in the window $W_{i,j}$, I_0 and σ_0 are the expected mean and standard deviation, respectively.

$$I'_{i,j} = \frac{\sigma_0}{\sigma_{W_{i,j}}}(I_{i,j} - \overline{I}_{W_{i,j}}) + I_0. \tag{1}$$

In order to represent the local information better, the size of the window should be smaller than the estimated size of the plate. In this paper, we select a 48×36 rectangle as the window $W_{i,j}$ and thus 8×8 windows can cover over the whole 384×288 car image. Let I_0 be equal to $\overline{I}_{W_{i,j}}$ and σ_0 be a constant independent of pixel $P_{i,j}$. Now we need to know the values $\overline{I}_{W_{i,j}}$ and $\sigma_{W_{i,j}}$ at each pixel. Computing out all the values is not advisable, and we can use the bilinear interpolation algorithm to get them. First we cut the car image into 8×8 blocks equably; and then compute out the $\overline{I}_{W_{i,j}}$ and $\sigma_{W_{i,j}}$ values at the vertexes of blocks, where $i = 36m$, $j = 48n, m, n = 0, 1, 2, \ldots, 8$; finally compute out every $\overline{I}_{W_{i,j}}$ and $\sigma_{W_{i,j}}$ by the bilinear interpolation Eqs. (2) and (3) (Fig. 2), where $36m \leqslant i < 36(m+1)$, $48n \leqslant j < 48(n+1)$, $c_x = (j - 48n)/48$, and $c_y = (i - 36m)/36$.

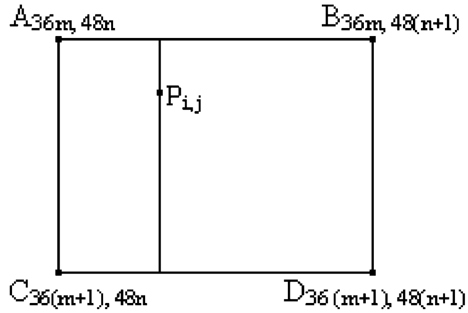

Fig. 2. Compute the $\sigma_{W_{i,j}}$ value in the rectangle block ABCD by bilinear interpolation.

$$\begin{aligned}\overline{I}_{W_{i,j}} &= (1 - c_y)[(1 - c_x)\overline{I}_{W_A} + c_x\overline{I}_{W_B}] \\ &\quad + c_y[(1 - c_x)\overline{I}_{W_C} + c_x\overline{I}_{W_D}],\end{aligned} \tag{2}$$

$$\begin{aligned}\sigma_{W_{i,j}} &= (1 - c_y)[(1 - c_x)\sigma_{W_A} + c_x\sigma_{W_B}] \\ &\quad + c_y[(1 - c_x)\sigma_{W_C} + c_x\sigma_{W_D}].\end{aligned} \tag{3}$$

If $\sigma_{W_{i,j}}$ is close to zero (such as only a dark or bright area), Eq. (1) will give out a large value. But we should not enhance such a local area. If $\sigma_{W_{i,j}}$ is high enough (for example $\sigma_{W_{i,j}} \geqslant 60$), the enhancement is unnecessary too. So the enhancement Eq. (1) is improved into Eq. (4) in practice.

$$I'_{i,j} = f(\sigma_{W_{i,j}}) \cdot (I_{i,j} - \overline{I}_{W_{i,j}}) + \overline{I}_{W_{i,j}}, \tag{4}$$

where $f(\sigma_{W_{i,j}})$ is an enhancement coefficient (shown in Fig. 3) defined by Eq. (5). Most $\sigma_{W_{i,j}}$s of the plate areas which need enhanced are around 20. So we let the function f be equal to 1 when $\sigma_{W_{i,j}} = 0$ or $\sigma_{W_{i,j}} \geqslant 60$, and be equal to 3 (as $20 \times 3 = 60$) when $\sigma_{W_{i,j}} = 20$.

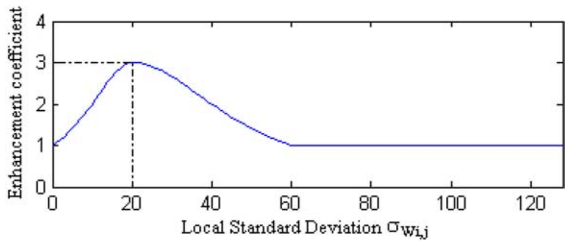

Fig. 3. The enhancement coefficient $f(\sigma_{W_{i,j}})$.

Fig. 4. The enhanced car image I'.

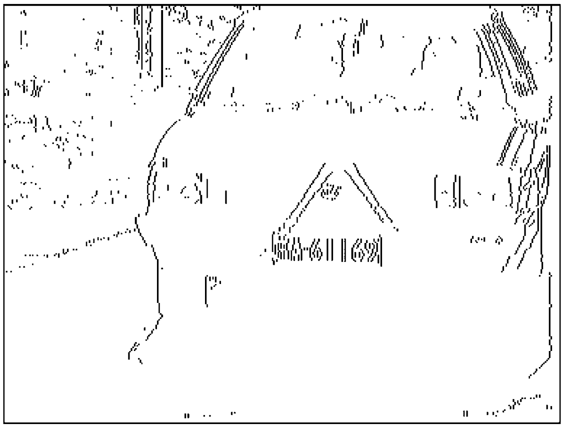

Fig. 6. The vertical edge image.

$$f(\sigma_{W_{i,j}}) = \begin{cases} \dfrac{3}{\dfrac{2}{400}(\sigma_{W_{i,j}} - 20)^2 + 1} & \text{if } 0 \leqslant \sigma_{W_{i,j}} < 20. \\ \dfrac{3}{\dfrac{2}{1600}(\sigma_{W_{i,j}} - 20)^2 + 1} & \text{if } 20 \leqslant \sigma_{W_{i,j}} < 60. \\ 1 & \text{if } \sigma_{W_{i,j}} \geqslant 60. \end{cases} \tag{5}$$

The enhanced car image is shown in Fig. 4. And we can see that the license plate region has been strengthened. If the plate is well illuminated and the image is in balance, the process will not change the contrast of the plate ($f(\sigma_{W_{i,j}}) = 1$, if $\sigma_{W_{i,j}} \geqslant 60$).

2.2. Vertical edge extraction

We select the vertical Sobel operator (in Fig. 5) to detect the vertical edges, because the simple operator costs us a little computational time. Convolve the car image with this Sobel operator to get the vertical gradient image. Compute the mean of the absolute gradient values in the image and multiply it by a coefficient as a threshold (for example $4\overline{Grad}$), or compute the gradient histogram and find a gradient at a certain percentage (for example 75%) of the gradient distribution as a threshold. Use this threshold and apply nonmaximum suppression in horizontal direction in the gradient image, and we get the vertical Sobel edge image shown in Fig. 6.

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

Fig. 5. The vertical Sobel operator.

2.3. Background curve and noise removing

From Fig. 6, we can see that there are many long background curves and short random noise edges in the vertical edge image besides the license plate edges. These background and noise edges may interfere in the license plate location. We have proposed a simple algorithm to remove them from the edge image.

This algorithm only requires us to scan the edge image for three times. The first scan will record the edge lengths away from the top (or left) start points. And the second scan will record the edge lengths away from the bottom (or right) end points. And the last scan will add up the two kinds of lengths to denote the actual edge lengths; if the edge point has a very long (background curve) or very short (noise edge) actual edge length, then the edge point will be removed from the edge image. Before describe the algorithm below, we need to introduce some symbols used: E denotes the edge image (if pixel $P_{i,j}$ is an edge point, $E_{i,j} = 1$, else $E_{i,j} = 0$); M and N are both matrixes with the same size as E; T_{long}, which has relation to the estimated height of the license plate, and T_{short}, which is shorter than most of the lengths of the plate edges, are two thresholds of edge lengths.

1. initialize M and N to zero matrixes;
2. for each row i from top-to-bottom do
    for each column j from left-to-right do
      if ($E_{i,j} == 1$)
        if ($E_{i-1,j-1} + E_{i-1,j} + E_{i-1,j+1} + E_{i,j-1} > 0$)
          $M_{i,j} = \max\{M_{i-1,j-1}, M_{i-1,j}, M_{i-1,j+1}, M_{i,j-1}\} + 1$;
        else
          $M_{i,j} = \max\{M_{i-2,j-1}, M_{i-2,j}, M_{i-2,j+1}, M_{i-1,j-2}, M_{i-1,j+2}, M_{i,j-2}\} + 1$;
        end
      end
    end
  end
3. for each row i from bottom-to-top do
    for each column j from right-to-left do
      if ($E_{i,j} == 1$)
        if ($E_{i+1,j-1} + E_{i+1,j} + E_{i+1,j+1} + E_{i,j+1} > 0$)
          $N_{i,j} = \max\{N_{i+1,j-1}, N_{i+1,j}, N_{i+1,j+1}, N_{i,j+1}\} + 1$;
        else
          $N_{i,j} = \max\{N_{i+2,j-1}, N_{i+2,j}, N_{i+2,j+1}, N_{i+1,j-2}, N_{i+1,j+2}, N_{i,j+2}\} + 1$;
        end
      end
    end
  end
4. for each row i from top-to-bottom do
    for each column j from left-to-right do
      if ($E_{i,j} == 1$)
        if ($M_{i,j} + N_{i,j} > T_{\text{long}} \| M_{i,j} + N_{i,j} < T_{\text{short}}$)
          $E_{i,j} = 0$;
        end
      end
    end
  end

In the above algorithm, we accumulate the edge lengths through observing the "concerned neighborhood pixels" (CNP) of the current pixel $P_{i,j}$. Fig. 7 shows the CNP in shadow grids.

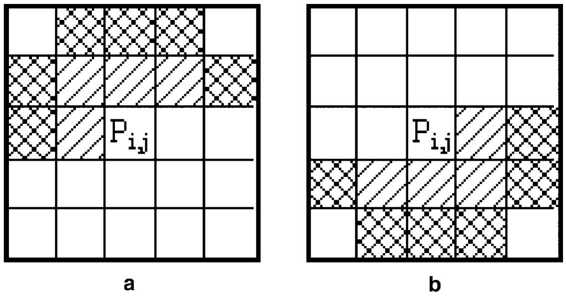

Fig. 7. The CNP of the pixel $P_{i,j}$: (a) scan image from left-to-right and top-to-bottom; (b) scan image from right-to-left and bottom-to-top.

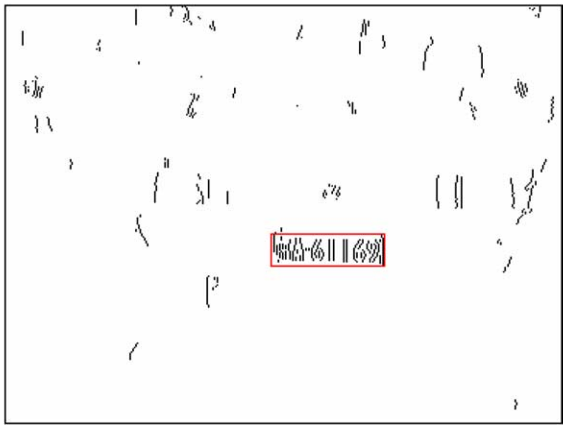

Fig. 8. The edge image after removing background and noise edges ($T_{\text{long}} = 28$, $T_{\text{short}} = 5$).

The result (Fig. 8) shows that most of the background and noise edges have been eliminated, but the license plate edges are almost fully saved.

2.4. License plate search and segmentation

After non-plate edges have been heavily removed, license plate location becomes much easier. We can shift a rectangle window whose size is just bigger than that of the license plate (for example 80×32) from left-to-right and top-to-bottom in the edge image. Count the total number of the edge points in the window. If the number is above a certain percentage of the area of the window, there may be a license plate in the corresponding window.

a b

Fig. 9. (a) The 48×36 convolution output image B' and (b) the license plate located from the car image in Fig. 1.

In order to expedite the search process, we let the window shifted by some X_{step} and Y_{step} (for example $X_{\text{step}} = 8, Y_{\text{step}} = 8$) instead of by pixel. And the following four steps tell how to locate the license plate:

- Cut the 384×288 edge image into many 8×8 blocks equably, and count the number of edge points in each little block, and form a 48×36 block image B.
- Use a $(80/8) \times (32/8) = 10 \times 4$ matrix W (each element is equal to 1) to denote the window. Convolve the image B with the window W, and export the image B'.
- If $B'_{i,j}$ is above the threshold T_{plate} and $B'_{i,j}$ is the local maximum, then record position $P_{8i,8j}$ as one of the plate candidates.
- Search all the candidates in their local regions, sort them by their B' values and segment them out from the original car image.

The convolution result $B' = B \otimes W$ is shown in Fig. 9(a). The response in the plate center is far higher than the other areas, so only one candidate will be searched in general. The plate segmented from the car image in Fig. 1 is shown in Fig. 9(b). The rest tasks are plate distortion correction, characters cutting and characters recognition, but they are not discussed in this paper.

3. Experimental results

In this section we compare the performance of the proposed method against some other used methods: “line sensitive filters” (Luis et al., 1999), “row-wise and column-wise DFTs” (Parisi et al., 1998), and “edge image improvement” (Ming et al., 1996). The “vector quantization” (Rodolfo and Stefano, 2000) is mainly used for image coding; and “color feature” (Wei et al., 2001; Zhu et al., 2002) is not robust enough for weather conditions, extra lights or dirty plates; and “fuzzy logic approach” (Zimic et al., 1997) works well under the assumption that the majority of plates are white with black characters, while most of Chinese license plates are blue with white characters. Therefore, these three methods were not employed in our comparative experiments.

The “line sensitive filters” method consists of three steps: subsampling image, applying line sensitive filters, and looking for rectangle regions containing maxima of response. The “row-wise and column-wise DFTs” method involves four steps: decomposing expected harmonics by using horizontal DFT on the image, averaging the harmonics in the spatial frequency domain, finding the horizontal stripe of the image containing the plate by maximizing the energy, and finding the vertical position of the plate in the same way by using vertical DFT on the candidate stripes. And the “edge image improvement” method contains five steps: extracting the edge image using Sobel operators, computing the horizontal projections of the edge image, calculating the medium range of the edge density level, eliminating the highest and lowest portion of the horizontal projections to simplify the whole image, and finding the candidates of license plates.

Three sets of Chinese vehicle images were used in our experiments. The first set has 163 images, and they were captured on a gate of our campus. The second set has 218 images, and they were captured in the shadow of strong sunlight near a road. The third set has 784 images, in which there are many complex backgrounds such as trees, parked bicycles and so on, and these images were taken from morning till night.

We performed all the four methods on the three sets. The Table 1 shows the experimental results. For each image, we located $1 \sim 3$ plate candidates, and the numbers of license plates hit by the first, the second and the third candidates are listed in Table 1, respectively. From this table we can see that the proposed method outperforms much over the other three methods: most of the license plates

Table 1
Comparison of location rates among the four methods on the three sets

Methods	First candidates	Second candidates	Third candidates	Plates not detected	Location rates (%)
Line sensitive filters	88	47	21	7	95.7
	125	40	34	19	91.3
	305	176	167	136	82.7
Row-wise and column-wise DFTs	79	45	33	6	96.3
	98	87	21	12	94.5
	316	172	155	141	82.0
Edge image improvement	129	23	7	4	97.5
	182	22	4	10	95.4
	473	156	84	71	90.9
The proposed method	161	2	0	0	100
	218	0	0	0	100
	755	24	3	2	99.7

Table 2
Comparison of computational times among the four methods (unit: ms)

Line sensitive filters	Row-wise and column-wise DFTs	Edge image improvement	The proposed method
19.6	21.5	16.4	47.9

Table 3
The processing times for the four stages in the proposed method (unit: ms)

Image enhancement	Vertical edge extraction	Background and noise removing	Plate search and segmentation	Total time
26	6.8	5.7	9.4	47.9

are found by the first candidates, and the location rates are almost all 100% (The two missing plates in the third set are too small). And the high location rates on the three sets reveal the robustness and efficiency of our method in license plate location. For the other three methods, most mislocated plates happened with the images containing some special objects (brands, radiators, bumpers) or complex backgrounds (trees, bicycles), and the images captured against the strong sunlight or under the gloomy light.

The computational times of the four methods are shown in Table 2, when they run on a Pentium-4 2.4 GHz, 256 MB RAM PC. The proposed method is the slowest one among the four methods.

The average processing times for the four stages of the proposed method are listed in Table 3. A lot of the time is consumed on the first stage "image enhancement". The total time of processing one 384×288 image is 47.9 ms, and it meets the requirement of real time processing.

4. Conclusion

The proposed method of license plate location makes use of the rich edge information in the plate area. In Section 2.1, we enhance the local areas in the original car image, but it is an alternative to enhance the gradient image to intensify the texture of the plate region. To avoid the interference factors around the plate, only vertical edges are extracted in Section 2.2. If the vertical edges, the left diagonal edges and the right diagonal edges are all extracted, a better continuity in edge curves can be attained at the expense of more computation time. The survival isolated short edges in Fig. 8 remain to be eliminated but not necessary.

The method has still some drawbacks. In Section 2.1, the $\overline{I}_{W_{i,j}}$ and $\sigma_{W_{i,j}}$ calculated by bilinear interpolation algorithm are not the actual values at the point P_{ij}. The "integral image" algorithm can solve this problem, but it takes much computation time too. Some numbers in this paper are relative to the estimated size of the license plate.

So if all the license plates in the images have the same size, the method will work better.

A great effect of our method in license plate location has been confirmed by the experiments. And some of the ideas in our method can be applied to some other applications, such as "text extraction on brands, envelopes, cards, bills…", "object segmentation in complex background", etc.

References

Lim, B.L., Yeo, W.Z., Tan, K.Y., Teo, C.Y., 1998. A novel DSP based real-time character classification and recognition algorithm for car plate detection and recognition. In: Fourth Internat. Conf. on Signal Process., vol. 2, pp. 1269–1272.

Luis, S., Jose, M., Enrique, R., Narucuso, G., 1999. Automatic car plate detection and recognition through intelligent vision engineering. In: Proc. IEEE 33rd Internat. Carnahan Conf. on Security Tech., pp. 71–76.

Ming, G.H., Harvey, A.L., Danelutti, P., 1996. Car number plate detection with edge image improvement. In: Fourth Internat. Symp. on Signal Process. and its Applications, vol. 2, pp. 597–600.

Parisi, R., Claudio, E.D., Lucarelli, G., Orlandi, G., 1998. Car plate recognition by neural networks and image processing. In: Proc. IEEE Internat. Symp. on Circuits and Systems, vol. 3, pp. 195–198.

Park, S.H., Kim, K.I., Jung, K., Kim, H.J., 1999. Locating car license plate using neural networks. Electron. Lett. 35 (17), 1475–1477.

Rodolfo, Z., Stefano, R., 2000. Vector quantization for license plate location and image coding. IEEE Trans. Industrial Electron. 47 (1), 159–167.

Wei, W., Wang, M.J., Huang, Z.X., 2001. An automatic method of location for number-plate using color features. In: Proc. Internat. Conf. on Image Process., vol. 1, pp. 782–785.

Zimic, N., Ficzko, J., Mraz, M., Virant, J., 1997. The fuzzy logic approach to car number plate locating problem. In: Proc. Intelligent Information Systems, pp. 227–230.

Zhu, W.G., Hou, G.J., Jia, X., 2002. A study of locating vehicle license plate based on color feature and mathematical morphology. In: 6th Internat. Conf. on Signal Process., vol. 1, pp. 748–751.