1999 第十二屆電腦視覺、圖學暨影像處理研討會，台北，中華民國八十八年八月廿二至廿四日，728—733 頁。

Efficiently Locating Vehicle License Plates Based on Vertical Line Detection

Wen-Bing Horng[†](洪文斌) Mong-Tsung Wu[†](吳孟璁) Chun-Hai Fan[‡](范俊海)
[†]Department of Computer Science and Information Engineering
[‡]Department of Transportation Management
Tamkang University
Taipei, Tamsui 25137, Taiwan, ROC
E-mail: {horng, chunhai}@mail.tku.edu.tw

Abstract

A fast method for locating vehicle license plates is proposed in this paper. This approach is primarily based on the observations that the intensity contrast between license plate background color and symbol color is high and that symbol edges are clustered in the plate area. A simple edge detection method based on first derivatives is applied to a car image such that the binary image containing only vertical edges is obtained. Then possible plate regions could be located from the horizontal and vertical projections of the binary image. No preprocessing, such as image enhancement and noise filtering, is required. In addition to its simplicity and fastness, the proposed method is less sensitive to illumination changes of the plate. The proposed method was experimented with 117 diverse car images on a Pentium-133 processor. The system successfully located 116 license plates in 0.2 second on an average.

1. Introduction

Recently, vehicle license plate recognition has drawn a lot of attention due to its practical demands. These applications include parking lot management, automatic collection of tolls in highways, travel-time measurements, identification of stolen cars, control of restricted access areas, speed-limit enforcement, and so on. A license plate contains a Vehicle Identification Number (VIN), which usually consists of such symbols as digital numbers 0-9 and capital roman letters A-Z. A typical car image, usually represented as a gray scale, is shown in Figure 1, in which the VIN is KE-2418.

In general, a vehicle license plate recognition system could be divided into two phases: *segmentation* and *recognition*. The purpose of the segmentation phase is to isolate *glyphs* (images of individual symbols) in the VIN from a car image. Then, each extracted glyph is sent to the recognition phase for identifying the VIN. As indicated in [12], the recognition speed of the system is mainly dominated by the segmentation phase. Therefore, increasing the segmentation speed will improve the overall performance of the whole system significantly.

In this paper, a fast and effective method is proposed to locate license plate in order to accelerate the speed of the segmentation phase in the vehicle license plate recognition system. This method is primarily based on the observations that the intensity contrast between license plate background color and VIN symbol color is high and that symbol edges are clustered in the plate area. An original car image is converted into a binary image containing only vertical edges through a simple edge detection method based on first derivatives. Then possible plate regions could be detected by the horizontal and vertical projections of the binary image. No preprocessing, such as image enhancement and noise filtering, is required. In addition to its simplicity and fastness, the proposed method is less sensitive to illumination changes of the plate. The proposed method was experimented with 117 diverse car images on a Pentium-133 processor. The system successfully located 116 license plates in 0.2 second on an average.

Figure 1. A typical car image.

2. Literature survey

In order to successfully extract glyphs in the license plate of a car image, the following observations could be very helpful:

- The local contrast between license plate background and symbol color is high.
- The transition between background and foreground colors is repeatedly occurred.
- VIN symbol edges are clustered in the license

1999 第十二屆電腦視覺、圖學暨影像處理研討會，台北，中華民國八十八年八月廿二至廿四日，728—733 頁。

plate area.

There are two major approaches for isolating VIN glyphs in the literature. One approach is to directly extract all possible glyphs from an original car image. The other approach is to locate the license plate area in the car image first, and then to extract VIN glyphs.

In the first approach, the common preprocessing step is to convert an input car image into black and white by applying a global threshold technique such as the Otsu's method [15] or the Tsai's moment-preserving method [20]. Then, different methods are used to isolate possible glyphs. Chow [3] utilized the connected component concept to extract glyphs. Cowell [5] employed the region growing technique to remove background so that only the VIN glyphs will be remained. This approach suffers from one major difficulty: Global threshold techniques might lose some detailed information in the binary plate image [18]. As shown in Figure 2, the first row contains two input car images. The second and third rows show their binary images resulted from the Otsu's and the Tsai's moment-preserving thresholding methods, respectively. This is due to changeable and uneven illumination caused by dirt and shadows on the license plate, changes in weather conditions, or deformation of the license plate.

Figure 2. Global thresholding of car images.

The second approach of isolating VIN glyphs is to locate the license plate in an input car image first. Choi [2] and Kim [9] proposed a method based on vertical edge detection using Hough Transform to extract license plates. This is assumed that only license plates have vertical edges in front image of a vehicle. However, this method has several drawbacks: (i) many vehicle images have vertical edges from radiators, (ii) Hough Transform is very sensitive to deformation of a plate boundary, and (iii) it needs much computation time. Nagano et al. [13] based on local statistics and template matching to select possible plate regions. However, template matching might not be efficient when the window moving around the image. Lee et al. [11] employed a neural network for color extraction to segment license plates. However, in addition to time-consuming, the method might fail if the vehicle body and the license plate background have similar colors. Soh et al. [19] proposed a robust method based on gray level profiles of horizontal lines to search license plate regions. Nijhuis et al. [14] used fuzzy c-means clustering algorithms to extract plate areas. Comelli et al. [4] analyzed gradients of the image to estimate the centers of license plates. Kim et al. [10] utilized a distributed genetic algorithm to extract plate regions on color images. Cui and Huang [6] employed spatial variances to locate text regions in plates. Draghici [7] and Busch et al. [1] applied Sobel operator for vertical and horizontal edges to find plate areas. ter Brugge et al. [21] used discrete-time cellular neural networks for extracting license plate regions. Parisi et al. [16] applied 1-D Discrete Fourier Transform scheme to find plate areas.

3. Basic notation

Before delving into the method of locating license plates in a car image, let us see what a license plate is and some basic notation used in this discussion first. , As shown in Figure 3, a typical vehicle license plate in Taiwan is 32 cm wide and 15 cm high. Each symbol it contains is 4.5 cm wide and 9.0 cm high. It mainly contains a VIN, which consists of a capital letter followed by a capital letter or a digit followed by a small dash and four digits. That is, a VIN contains six symbols (including A-Z and 0-9) and a small dash, with the dash separating the six symbols into two parts. The first part contains two symbols and second part consists of four digital symbols only. Above the VIN there are three Chinese characters indicating the place where the car is registered.

Figure 3. A typical vehicle license plate.

Let N be the set of natural numbers, (x, y) be the spatial coordinate of a digitized image, and $G = \{0, 1, \ldots, l-1\}$ be a set of positive integer representation gray levels. Then, an *image function* can be defined as the mapping $f : N \times N \rightarrow G$. The brightness (i.e., gray level) of a pixel with coordinate (x, y) is denoted as $f(x, y)$. The origin is at the lower-left corner of an image with the x-axis horizontal and the y-axis vertical.

Let $t \in G$ be a threshold, $B = \{b_0, b_1\}$ be a pair of binary gray levels, and $b_0, b_1 \in G$. The result of *thresholding* an image function $f(\cdot, \cdot)$ at gray level t is a *binary image* function $f_t : N \times N \rightarrow B$, such that $f_t(x, y) = b_0$ if $f(x, y) < t$, and b_1 otherwise.

The gray level *histogram* of an image function with gray levels in G is a discrete function $h_g : G \rightarrow N$, such that $h_g(r_k) = n_k$, where r_k is the kth gray level, n_k is the number of pixels in the image with that gray level, and $k = 0, 1, 2, \ldots, l-1$. Each $h_g(r_k)$ is called a *bin* of the histogram. The *vertical projection* of a binary image function is a discrete function $h_x : N \rightarrow N$, such that $h_x(x) = n_x$, where x is an x-axis coordinate, and n_x is the number of pixels at x-axis coordinate x in the binary image with gray level b_0. The *horizontal projection* of a binary image function is a discrete function $h_y : N \rightarrow N$, such that $h_y(y) = n_y$, where y is an y-axis coordinate, and n_y is the number of pixels at y-axis coordinate y in the binary image with gray level b_0.

In this discussion, a car image has 640×480 pixels with 256 gray levels from 0 to 255. Gray level 0 is used as black color and gray level 255 is used as white color.

4. Locating license plates

As discussed in Section 2, the proposed method for locating license plates is primarily based on the observations that the local contrast between license plate background and VIN symbols is high and that symbol edges are grouped in the plate region. Thus, edge detection will be the first step to solve the problem. Once, the edges are detected, possible plate regions could be found based on the horizontal and vertical projections of the binary image. For the convenience of discussion, the proposed method could be divided into six steps, which are detailed in the following.

Step 1: Detect vertical edges

In order to accelerate the process and to reduce the amount of information, only vertical edges are detected in the proposed method since vertical edges of plate symbols contain more information than horizontal edges. In this method, a simple vertical edge detection method based on first derivatives [8][17] is used as follows:

```
for each pair of neighbor pixels in a horizontal scan line
  if the gray level difference is greater than some
       threshold
  then set the latter pixel to black
  else set the latter pixel to white
```

Figure 4 shows the result of this simple edge detection method with threshold 50 obtained from Figure 1.

Step 2: Compute the horizontal projection

Once the vertical edges are detected, the horizontal projection, hproj[ ], is computed as follows:

```
for each pixel in the horizontal scan line y
  if the pixel is black
  then increment hproj[y]
```

Figure 5 depicts the computed horizontal projection on the left of the image.

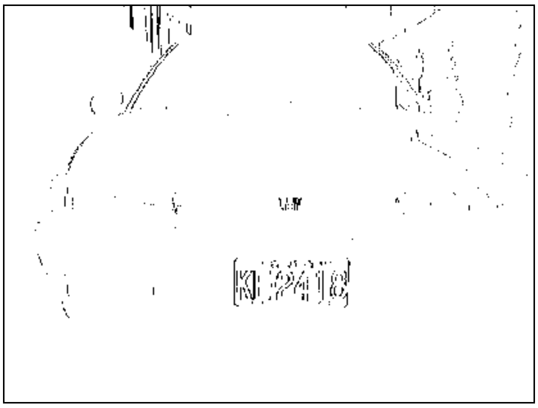

Figure 4. Vertical edge detection resulted from Figure 1 with threshold 50.

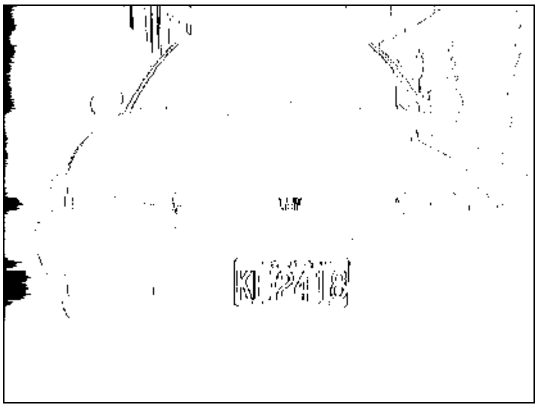

Figure 5. The horizontal projection.

Step 3: Find candidate y-ranges of the plate

Observing from the above horizontal projection, it is easy to see that the corresponding bins of the plate region in the projection are relative higher than other bins. This phenomenon justifies the previous observation that edges will be clustered on the plate region. Then, the problem becomes how to choose a suitable range on the y-axis (called *y-range*) for the plate. Recall that a plate VIN contains six symbols in this discussion. If all the edges were detected, there would be at least 12 black pixels for each horizontal scan line in the plate region (e.g., if the VIN is II-1111). Thus, each y-range satisfied this condition would be a possible y-axis range for the license plate. Figure 6 illustrates this selection process. A vertical line at $x = 12$ is drawn on the left such that each bin in the y-axis projection with yproj[y] < 12 is not shown in the figure. However, not every possible y-ranges are reasonable, because too small symbol images may cause recognition difficulty. It is also rarely that a plate occupies a large portion of the image. Therefore, a reasonable range [minheight, maxheight] is predefined such that each detected *y*-range whose height is within this range is a candidate *y*-range for the following process.

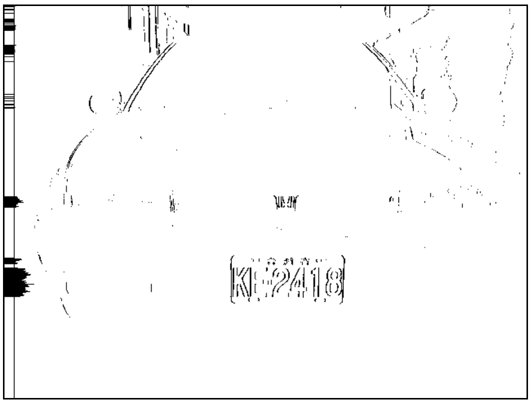

Figure 6. Possible *y*-ranges for the license plate.

Step 4: Compute the vertical projection

The computation of the vertical projection, vproj[ ], for a candidate *y*-range is easily achieved as follows:

```
for each pixel with coordinate (x, y) in the horizontal line
        within an y-range
   if the pixel is black
   then increment vproj[x]
```

Step 5: Find candidate regions of the plate

Similarly, since edges are normally clustered in the plate area, the corresponding bins of the plate region in the vertical projection are usually relative higher than other bins. A suitable *x*-axis range (called *x-range*) could be selected from the above observation. However, since only vertical edges are detected in this method, the corresponding bins within the plate region are normally discretely distributed. That is, the values of these bins may not all be greater than zero. Since the width to height ratio of a symbol is 1:2, it is reasonable to consider two non-zero neighbor bins being *connected* if their distance is less than symbol width. (The corresponding *y*-range is reasonably assumed as symbol height.) However, there may exist noises (the black pixels other than those on the actual symbol edges) during the edge detection process. So, each bin whose value is less than a predefined threshold is reasonably regarded as a noise bin and its bin value is reset to zero. Once the vertical projection is updated, it is easily to find all possible *x*-ranges from the connected bins. Similarly, not every possible *x*-ranges are reasonable, since the width to height ratio of a standard plate is fixed. Therefore, the block enclosed by the *x*-range and *y*-range with a suitable width to height ratio is considered as a possible plate region. Figure 7 shows the only one possible plate region found in this step. The part of the vertical projection for the plate region is also shown in the figure.

Step 6: Validate each possible plate region

As stated previously, there may exist several other regions whose properties are similar to those of the license plate, such as front-grill, marks, or other symbols. Therefore, some conditions must be set to check whether each possible plate region is a candidate region. Some conditions used in this paper are listed as follows:

- The number of black pixels in the possible plate region should be greater than a multiple of block height.
- From the possible plate region in the binary image, construct its horizontal projection. The number of bins with their values greater 12 should have at least a minimum ratio of the block height.

The plate region found in Figure 7 is validated as a candidate plate region for the isolation of symbol images in the segmentation phase. If there are no candidate plate regions selected in this process (from Step 1 to Step 6), called a *trial*, go to Step 1 with a smaller threshold for edge detection. This might be caused by low contrast between plate background and symbol color owing to dirt or low illumination.

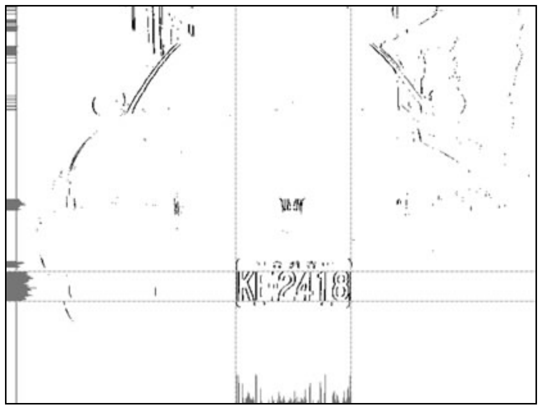

Figure 7. The found plate region.

5. Experimental Results

The proposed method for locating vehicle license plates was experimented with 117 diverse car images on a Pentium-133 processor. The experiment starts with the threshold 50 for edge detection in Step 1. If no candidate plate regions are found, then decrement the threshold by 10 until it gets to 10.

For the 117 test car images, 115 of them successfully find only one candidate plate region, in which 114 candidate regions are correct blocks of the plates and the other one is mistaken (refer to Figure 8). Each of the remaining two test images has 2 candidate plate regions, and in each case one of the candidate regions is correctly identified (refer to Figure 9). The experimental result is shown in Table 1. Therefore, the identification rate for the set of test images is 116/117 = 99.14%. Recall that each of the above process (from Step 1 to Step 6) for locating candidate

1999 第十二屆電腦視覺、圖學暨影像處理研討會，台北，中華民國八十八年八月廿二至廿四日，728—733 頁。

plate regions is called a trial. Table 2 shows the number of trials for the test car images. There are 86 car images with only one trial. The average number of trials for the test images is 1.39, and each trial takes 0.14 second in this experiment. Therefore, the system could locate the license plate from a car image in 0.2 second on an average.

Table 1: Number of candidate regions

Candidate regions	Images	Correct
1	115	114
2	2	2

Table 2: Number of trials for test car images

Threshold	Trials	Images
50	1	86
40	2	22
30	3	5
20	4	2
10	5	2
Average trials:	1.39	

Figure 8. Wrong license plate region.

Figure 9. Two candidate license plate regions.

6. Conclusion

In this paper a fast method for locating vehicle license plates is proposed. It is mainly based on the observations that the intensity contrast between license plate background color and symbol color is high and that symbol edges are clustered in the plate area. Possible plate regions could be found from a simple vertical edge detection and the inspection of the horizontal and vertical projections of the binary edge image. In addition to its simplicity and fastness, the proposed method is less sensitive to illumination changes of the plate. The proposed method is tested with 117 car images, and 116 of them can successfully find correct plate regions. Thus, 99% identification rate is achieved for the test images. An average of 1.39 trials is used to locate the plate region with each trial taking 0.14 second in the experiment on a Pentium-133 processor. On an average, the system could locate the license plate from a car image in 0.2 second. Thus, fastness and high identification rate for locating vehicle license plates are achieved by the proposed method.

References

[1] C. Busch, R. Dorner, C. Freytag, and H. Ziegler, "Feature Based Recognition of Traffic Video Streams for Online Route Tracing," in *Proceedings of the 48th IEEE Vehicular Technology Conference* (VTC 98), Vol. 3, 1998, pp. 1790–1794.

[2] H.J. Choi, "A Study on the Extraction and Recognition of a Car Number Plate by Image Processing," *Journal of the Korea Institute of Telematics and Electronics*, Vol. 24, 1987, pp. 309–315.

[3] J.N. Chow, *License Plate Recognition System*, Master Thesis, National Sun Yat-Sen University, Taiwan, June 1995.

[4] P. Comelli, P. Ferragina, M.N. Granieri, and F. Stabile, "Optical Recognition of Motor Vehicle License Plates," *IEEE Transactions on Vehicular Technology*, Vol. 44, No. 4, November 1995, pp. 790–799.

[5] J.R. Cowell, "Syntactic Pattern Recognizer for Vehicle Identification Numbers," *Image and Vision Computing*, Vol. 13, No. 1, 1995, pp. 13–19.

[6] Y. Cui and Q. Huang, "Character Extraction of License Plates from Video," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 17–19, 1997, pp. 502–507.

[7] S. Draghici, "A Neural Network Based Artificial Vision System for License Plate Recognition," *International Journal of Neural Systems*, Vol. 8, No. 1, February 1997, pp. 113–126.

[8] R.C. Gonzalez and R.E. Woods, *Digital Image Processing*, Addison-Wesley, 1992.

[9] H.S. Kim, et al, "Recognition of a car number plate by a neural network," in *Proceedings of the Korea Information Science Society Fall Conference*, Vol. 18, No. 2, 1991, pp. 259–262.

[10] S.K. Kim, D.W. Kim, and H.J. Kim, "A Recognition of Vehicle License Plate Using a Genetic Algorithm Based Segmentation," in *Proceedings of the Third IEEE International Conference on Image Processing*, Vol. 2, 1996, pp. 661–664.

[11] E.R. Lee, P.K. Kim, and H.J. Kim, "Automatic Recognition of a Car License Plate Using Color Image Processing," in *Proceedings of the International Conference on Image Processing (ICIP'94)*, 1994, Vol. 2, pp. 301–305.

[12] F. Lisa, J. Carrabina, C. Perez-Vicente, N. Avellana, and E. Valderrama, "Two-bit Weights Are Enough to Solve Vehicle License Recognition Problem," in *Proceedings of 1993 IEEE International Conference on Neural Networks*, San Francisco, California, 1993, pp. 1242–1246.

[13] K. Nagano, M. Tamagawa, K. Miyamota, Y. Iida, and I. Fujita, "Application of Image Processing and Pattern Recognition—Vehicle License Plate Recognition," *Mitsubishi Technical Review*, Vol. 29, No. 2, 1992, pp. 77–80.

[14] J.A.G. Nijhuis, M.H. ter Brugge, and K.A. Helmholt, J.P.W. Pluim, L. Spaanenburg, R.S. Venema, and M.A. Westenberg, “Car License Plate Recognition with Neural Networks and Fuzzy Logic,” in *Proceedings of the IEEE International Conference on Neural Networks* (ICNN’95), Vol. 5, November 1995, pp. 2232–2236.

[15] N. Otsu, “A Threshold Selection Method from Gray-level Histogram,” *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-9, 1979, pp. 62–66.

[16] R. Parisi, E.D. Di Claudio, G. Lucarelli, and G. Orlandi, “Car Plate Recognition by Neural Networks and Image Processing,” in *Proceedings of the 1998 IEEE International Symposium on Circuits and Systems* (ISCAS’98), Vol. III, pp. 195–198.

[17] J.R. Parker, *Algorithms for Image Processing and Computer Vision*, John Wiely & Sons, 1997.

[18] P.K. Sahoo, S. Soltani, A.K.C. Wong, and Y.C. Chen, “A Survey of Thresholding Techniques,” *Computer Vision, Graphics, and Image Processing*, Vol. 41, 1988, pp. 233–260.

[19] Y.S. Soh, B.T. Chun, and H.S. Yoon, “Design of Real Time Vehicle Identification System,” in *Proceedings of the 1994 IEEE International Conference on Systems, Man, and Cybernetics* (Human, Information, and Technology), Vol. 3, 1994, pp. 2147–2152.

[20] W.H. Tsai, “Moment-Preserving Thresholding: A New Approach,” *Computer Vision, Graphics, and Image Processing*, Vol. 29, 1985, pp. 377–393.

[21] M.H. ter Brugge, J.H. Stevens, J.A.G. Nijhuis, and L. Spaanenburg, “License Plate Recognition Using DTCNNs,” in *Proceedings of the 1998 Fifth IEEE International Workshop on Cellular Neural Networks and Their Applications*, London, England, 14–17 April, 1998, pp. 212–217.