International Journal of Automotive Technology, Vol. 10, No. 2, pp. 205–210 (2009)
DOI 10.1007/s12239–009–0024–2

1229–9138/2009/045–09

DYNAMIC PROGRAMMING-BASED METHOD FOR EXTRACTION OF LICENSE PLATE NUMBERS OF SPEEDING VEHICLES ON THE HIGHWAY

D.-J. KANG*

School of Mechanical Engineering, Pusan National University, Busan 609-735, Korea

(Received 28 November 2007; Revised 17 September 2008)

ABSTRACT–In the last decade, vehicle identification systems have become a central element in many applications involving traffic law enforcement and security enhancement, such as locating stolen cars, automatic toll management, and access control to secure areas. As a method of vehicle identification, license plate recognition (LPR) systems play an important role and a number of such techniques have been proposed. In this paper, we describe a method for segmenting the main numeric characters on a license plate by introducing dynamic programming (DP) that optimizes the functionality describing the distribution of the intervals between characters, the alignment of the characters, and the threshold difference used to extract the character blobs. The proposed method functions very rapidly by applying the bottom-up approach of the DP algorithm and also robustly by minimizing the use of environment-dependent image features such as color and edges.

KEY WORDS : License plate number extraction, Dynamic programming, Multiple thresholds, Image labeling

1. INTRODUCTION

Recently, automatic LPR systems have emerged as a central law enforcement tool in the United States, Europe, and East Asia, similar to vehicle application tendency of intelligent sensors such as machine vision and image processing system (Wu *et al*., 2007; Jung *et al*., 2007). Certain technologies are already available to law enforcement agencies for use under limited environmental conditions, and many companies are attempting to develop a robust and fast commercial system (Civica company page; Data-Works Plus company page).

In Korea, the use of LPR systems for over-speed traffic law enforcement is required on major highways in all areas of the country, and several research projects on this topic are currently in progress. LPR systems in Korea consist of three major parts: vehicle sensing, image acquisition, and character recognition. The vehicle sensing part detects the car passing by on the road with proximity or loop coil sensors buried in the road surface, and then sends the detection signal to an image acquisition system to trigger a video camera.

The image acquisition system receives the trigger signal from the sensing components and captures a vehicle image using a high speed shutter to reduce motion blur. The recognition segment of the system analyzes the captured image and recognizes numbers. Because the system must operate at night as well as under daylight conditions, the use of a triggered stroboscopic light is essential to ensure that the images acquired are valid, and environment-independent image features are therefore very important for improvements in the consistency of the algorithm.

A typical algorithm in the LPR technique consists of two major parts: a license plate locating part and a license number identification part. First, the conventional methods of LPR attempt to find the license plate by using image features such as the shape, color, symmetry, or height-to-width ratio (Nijhuis *et al*., 2005; Chang *et al*., 2004). Then, the characters and numbers on the plate are segmented from the plate by using blobs, lines, the distribution of the gradient of magnitude and direction, or the alignment between characters (Chang *et al*., 2004; Hontani and Koga, 2001). The segmented characters are finally recognized by categorization algorithms such as a neural network or a probabilistic classification (Duda, *et al*., 2001). Conventional methods use environment-dependent image features such as colors, edges, blobs, or pixel intensities. Specifically, the color information is a useful image feature that can be used to separate the license plate region from the background of the vehicle. Hence, the algorithm performance is very sensitive to changes in environmental conditions that affect the quality of image features. For example, well-conditioned lightness, clear color properties, and acquisition of a high resolution image of the license plate are required to successfully segment the license plate of the target vehicle. In this paper, license plate detection of speeding vehicles on the highway does not satisfy the required environmental conditions because the LPR system

**Corresponding author.* e-mail: djkang@pusan.ac.kr

must operate during all 24 hours in a day, under the variable outdoor environment of the highway.

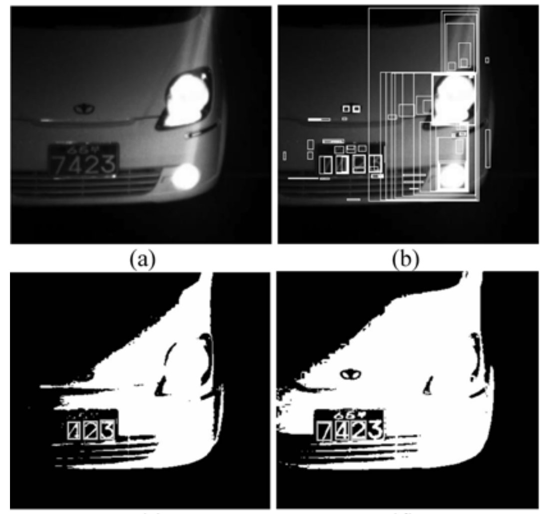
The method proposed in this paper does not require the plate location module, because it segments the numbers directly on the license plate. The proposed method also does not require any image features of the license plate such as edges, colors, or lines, which are always affected by intensity variations. Instead, the segmented blobs (after binarization of the gray-level image) are used to provide the geometric constraints for the numeric characters on the license plate.

First of all, a simple image binarization for the inputted gray level license image extracts many binary blobs in the target image. Because we do not know the optimal threshold value that will effectively distinguish between the background and characters of the license plate, an automatic threshold decision approach such as the Otzu method (Nixon and Aguado, 2002) is required, but the variation in the threshold value for image binarization is very large while the relative size of the license plate in the entire image region is very small. Therefore, typical automatic threshold methods do not provide a good threshold value for character segmentation under uncontrolled and dynamic illumination conditions on the highway. The purpose of image binarization is to separate character blobs from the background of the plate and other image clutter. Therefore, we try to extract the character blobs by artificially changing the threshold values over a wide range, from a small value to a large value. During the changes in the threshold value within a fixed range, correct character blobs can be segmented even though many binary blobs (including noisy clutter and background regions) are extracted. Among all of the candidate blobs obtained from the input images using a wide range of threshold values, those character blobs including four successively located numbers can be identified from the proposed dynamic programming framework.

2. EXTRACTION OF THE CANDIDATE REGIONS BY MULTIPLE THRESHOLDS

In all types of Korean vehicles, there are four large numbers in a single row in a defined region of the license plate. If we can extract the four numbers, then the position of the license plate can be identified and the other literals and numbers on the plate can be also segmented from the plate background region. Figure 1(a) shows the local part of a license plate image captured by a triggered stroboscopic light during nighttime operation. The intensity values of the four main numbers on the plate are not uniform, and we therefore cannot apply a single threshold value to separate the numbers from the gray-scale image.

Figures 1(c) and (d) show the two binary images obtained when threshold values of 70 and 40 are applied to the gray-scale image of Figure 1(a), respectively. In Figure 1(c), the numbers "4", "2", and "3" can be extracted by the



(a) (b) (c) (d)

Figure 1. Extraction of binary blob regions by changing the image threshold value.

image-labeling algorithm as separated blobs. The small boxes represent the minimum boundary rectangles (MBR) of the numeric character blobs. For the lower threshold value of 40 in Figure 1(d), another three numbers "7", "4" and "2" can be extracted, while the bottom part of the number "3" starts to blend with the background region. Because of the variation in blob extraction for the threshold change, the selection of a single threshold value by a fixed or adaptive threshold decision algorithm is very difficult. In this example, there is no single threshold value that allows the four numbers to be separated by binarization of the gray-scale image and use of a labeling algorithm. Figure 1(b) shows the MBRs for all of the binary blobs obtained during the change in the threshold value over a wide range. The properties of each blob include the height and width, the center position, and the threshold value used to extract the blob. They are used later as the geometric constraint to connect neighboring blobs.

Even though many noisy blobs of background appear, the correct blobs, including the four numbers, are also detected. Among all of the blobs considered as search candidates, the four numeric character blobs that are successively located can be extracted by means of an energy minimization framework using the geometric constraint between neighboring blobs.

3. ENERGY MODEL AND OPTIMIZATION

Assume that the four numbers of the license plate exist successively on a horizontal region of similar vertical position. The optimization process attempts to find an optimal configuration that minimizes the overall functional of the geometric constraint that the four number blobs can provide. While the horizontal distance between neighboring numbers should be maintained within the permitted range,

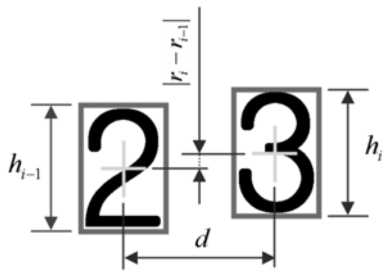


Figure 2. Geometric configuration between two neighboring numeric character blobs.

the vertical distance between the center positions of two neighboring numbers can be minimized. If two neighboring blobs with the center distance *d*, as shown in Figure 2, are within the permitted range, then they can be considered as correct candidates for part of the four numeric characters.

The energy model for the blobs of any successive two numbers consists of three constraint parts – two geometric constraints and a threshold constraint:

(1) The height of the two successive numbers must be similar:

$$E_h^i(v_i, v_{i-1})=\alpha \cdot |h(v_i) - h(v_{i-1})| \tag{1}$$

where $h(v_i)$ represents the height of node v_i considered as a candidate blob, and α is a weighting constant.

(2) The difference in the vertical distance between the center positions of the two neighboring numbers should be minimized:

$$E_r^i(v_i, v_{i-1})=\beta \cdot |r(v_i) - r(v_{i-1})|\,, \tag{2}$$

where $r(v_i)$ represents the row coordinate of the center position of node v_i.

(3) The threshold values applied to obtain the binary blobs for the two neighboring numbers should be similar to each other:

$$E_t^i(v_i, v_{i-1})=\gamma \cdot |t(v_i) - t(v_{i-1})|\,, \tag{3}$$

where $t(v_i)$ represents the threshold value used to obtain node v_i from the gray-scale image. This constraint requires that the intensity values of the numbers on a license plate should be similar to each other.

The total energy is then given by:

$$E_{total}=\sum_{i=1}^{N} [E_h^i(v_i, v_{i-1})+E_r^i(v_i, v_{i-1})+E_t^i(v_i, v_{i-1})] \tag{4}$$

This energy function can be minimized by the following recursive dynamic programming algorithm:

$$S(n,m)=\min_k \{S(n-1,k)+E_n(v_{n,m}, v_{n-1,k})\} \tag{5}$$

$$B(n,m)=\arg\min_k \{S(n-1,k)+E_n(v_{n,m}, v_{n-1,k})\}, \tag{6}$$

where *S*(*n*, *m*) is the accumulated minimal energy with increasing index, *n*, and *B*(*n*, *m*) holds the minimizing value of *k*(*k*=1, ..., *M*). The parameter *n*(*n*=1, ..., *N*) represents the index of successive numbers; for example, if we want to extract four numbers in a horizontal region, then $N=4$. The parameters $v_{n,m}$ and $v_{n-1,m}$ denote the current and previous blobs for two neighboring numbers, respectively. The index *m*(*m*=1, ..., *M*) presents a candidate blob among all of the saved blobs obtained from changing the threshold value, where *M* is the total number of candidate blobs. After all stages have been processed, an optimal path is obtained by backtracking, beginning with the candidate that minimizes *S*(*N*, *m*). The time cost of the algorithm is $O(N-1 \cdot M^2)$. As a bottom-up approach, dynamic programming provides a very fast solution to finding an optimal path on a graph (Neapolitan and Naimipour, 1998; Park and Kim, 2007). The DP method has the advantage of greatly reducing the time complexity for a candidate search based on local similarity. This aspect of the DP method is very useful for our algorithm. The pseudo-code of the dynamic programming for the energy function minimization is as follows:

for all *m*, *S*(1, *m*)=0
    for *n*=2, ..., *N*
        for *m*=1, ..., *M*

$$S(n,m)=\min_k \{S(n-1,k) + E_h(v_{n,m}, v_{n-1,k})+E_r(v_{n,m}, v_{n-1,k}) + E_t(v_{n,m}, v_{n-1,k})\}$$

$$B(n,m)=k^{\min}$$

4. EXPERIMENTS

In the experiments, we set the permitted range for the horizontal center distance, *d*, from 10 to 30 pixels. If the center distance of any two blobs corresponding to the main plate number exceeds the maximum value of 30 or is smaller than the minimum value of 10, the candidate node blob is eliminated from the list of search candidates of the current blob. The initial value of the threshold used to extract the binary blobs is set to 10 and then increases at intervals of 10 until it reaches 240 for an 8-bit gray-level image. The typical size of a license plate in the current experimental images is about 95×45 pixels, a low resolution in the whole image size of 692×516 pixels. For all of the images in the experiment, the weighting constants α, β, and γ in Equations (1)~(3) are set to 0.1, 0.1, and 0.01, respectively.

The test images were collected from over-speed cars that were detected by a loop coil sensor on the highway during the day and at night. Figure 3 shows an experimental result for a typical image set that include good plate images with desirable features such as clear and focused edges, good contrast between characters and background, and minimal intensity variation under proper lighting conditions. Figure 3(a) presents the detection result for day time and Figure 3(b) shows the result for night time. Both cases give good

(a)

(b)

Figure 3. Extraction of the four main numbers on the license plate for highway vehicles.

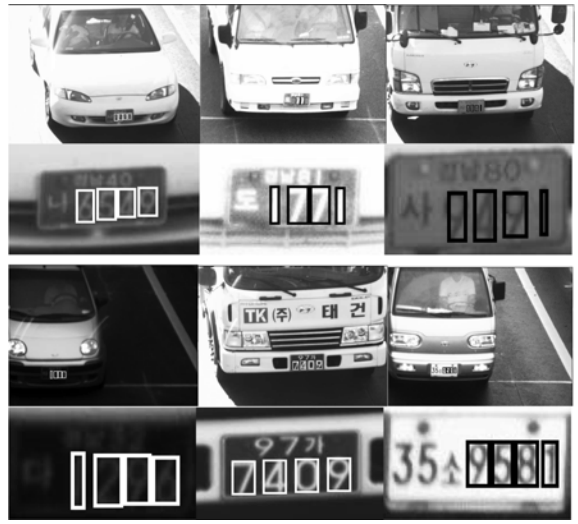


Figure 4. Experimental result for some difficult cases.

extraction results.

Table 1 summarizes the DP-based search algorithm presenting the optimal threshold values that were found and the computing time of each module. When the total number of binary blobs obtained during the wide variation of the threshold value increases, the computing time of the DP search also increases, but only slightly. The multiple values of the threshold, shown as bold in the third column of the table, indicate that there is no single optimal threshold value that can be used to segment the numbers on the license plate. Most of the computing time is occupied by the blob extraction module, because the image labeling algorithm is executed repetitively due to the change in the threshold value. The DP search algorithm is very fast, with a computing time of about 3 msec required for most cases in the experiment. However, the large number of binary blobs (418) in the fourth example needs about 6 msec because of the 2nd order dependency of the time complexity for the blob number.

Table 1. Performance summary for DP-based extraction system of license plate numbers. The number index presents the six images of Figure 3, respectively.

	# of blobs	Found threshold value	Computing time (sec)	
			Blob extraction	DP search
1	316	**110,120**	0.296	0.0037
2	333	120	0.281	0.0042
3	303	**80,90,100**	0.11	0.0018
4	418	100	0.218	0.0065
5	406	**90,100**	0.172	0.00532
6	256	**120,130**	0.141	0.0028

Because the proposed method uses repeated trials of threshold value for segmentation of characters on the license plate, most of the computing time for detecting the license plate is occupied by the blob coloring module. For reduction of computation and improvement of performance, applying automatic adaptive binarization methods such as those of Li or Sauvolar (Li *et al.*, 2005; Sauvolar *et al.*, 1997) may be very useful.

Figure 4 shows another experimental result from the proposed algorithm applied to some difficult cases, including an unfocused image, a blurred image with unclear character edges, a dark image at night, a diffuse reflection image during the daytime, and a noisy image with characters in the background. The numeric characters are also well segmented for all of these cases.

In our experiments, the total number of test images is 245, each with a size of 692×516 pixels, and the success rate for detection of the four main numbers is 97.14% with respect to correct interpretation of the license plate numbers made by a human user. The experiments were performed on a desktop computer equipped with an Intel Core-Duo 2.0GHz CPU.

The apparent shape of the license plate in an image depends on the relative location between the camera and the car. Even though the triggered camera is installed at a fixed position above the road surface, the location of the license plate in the image varies, with some skewed angles being observed, because the location of passing cars on the highway is arbitrary. The method is robust to the change of license plate position and angle within a limited range. In addition, the proposed energy model of the DP algorithm is not sensitive to the change of the license plate scale because the energy terms of Equation (1)~(3) do not directly use the size information for the numeric characters; instead, the geometric relation between two characters is used.

Figure 5 contains two images showing successful and

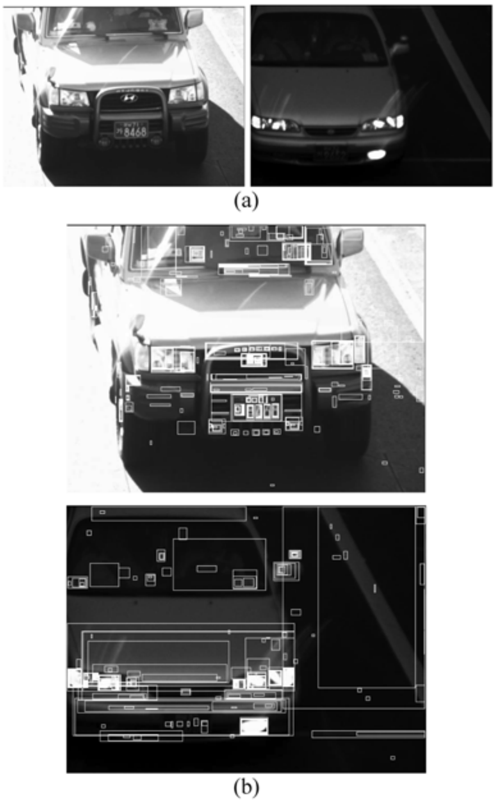


Figure 5. MBRs for two cases of successful and failed blob extraction.

failed extraction. In the first image of Figure 5(b), we can see the MBRs bounding the four main numbers on the license plate, but the MBRs of the second image of Figure 5(b) do not include the plate numbers and result in failure of the DP-based search because the candidate blobs cannot include the plate numbers. The most important criterion used to decide the success or failure of the algorithm is that of finding the optimal threshold value to extract the binary blobs for characters on the license plate. Therefore, we have to carefully control the apparent view using the strobe light and triggered camera to improve the image quality of the license plate. In this respect, the proposed method in this paper suggests a solution to the decision problem of threshold value in the case of severely changing dynamic illumination.

We compared our results with a method using the adaptive boosting (AdaBoost) algorithm (Dlagnekov, 2005; Zhang *et al*., 2006). The method uses a strong classifier as a combination of weak classifiers trained from the AdaBoost algorithm (Viola and Jones, 2004) and does not require prior segmentation of any character regions, such as the blobs, by image binarization or edge features. For each iteration, the method selects the best performing weak classifier from a set of weak classifiers, where each weak classifier acts on a single feature. After training, the strong classifier then combines the selected weak classifiers in a weighted manner. Owing to its powerful performance in finding objects of interest, the AdaBoost algorithm has been applied to detect human faces (Viola and Jones, 2004), text regions in road scenes (Chen and Yuille, 2004), walking people for video surveillance (Lee and Nevatia, 2007), and car license plates (Dlagnekov, 2005; Zhang *et al*., 2006).

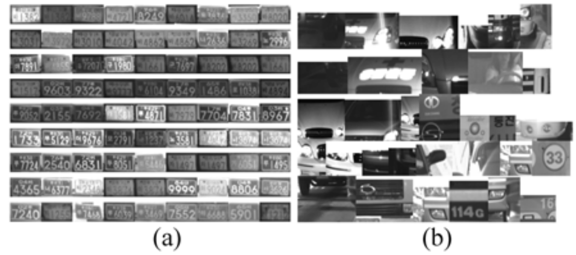


Figure 6. Positive and negative image samples for training the AdaBoot algorithm.

We used Intel OpenCV (Intel Company Page, 2008) to train the AdaBoost algorithm and find objects of interest. A total of 1,046 positive license plates and 5,524 negative images are used to train the strong classifier. Figure 6 shows a few samples from the training set, including positive and negative plate images. The training images were obtained from license plate images of speeding vehicles on a highway. The size and intensity variation in the training images are normalized and then inputted to train the AdaBoost classifier. After training, the classifier is applied to detect the license plates of vehicles on the highway. Figure 7 shows results of AdaBoot detection for highway vehicle images. The arrows in each image denote the area detected by the algorithm. Dlagnekov reported a detection rate of about 95% for car images in a parking lot (Dlagnekov, 2005), but our experiments with OpenCV AdaBoost showed a detection rate of only 62.5% for speeding cars on a highway, even though multiple detections of true and false areas were treated as correct recognition of the license plate. During both day and night observations, the license plate images are very noisy due to intensity variation and

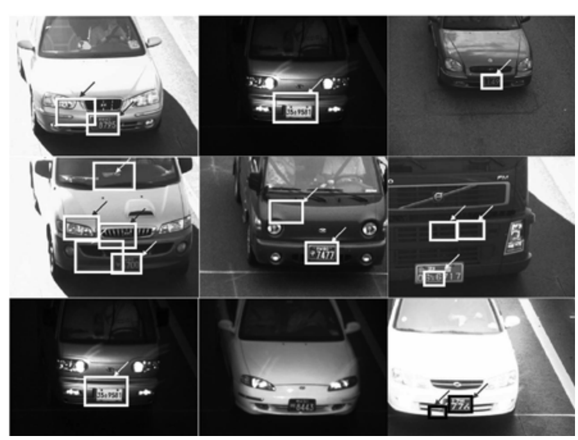


Figure 7. Recognition of the license plate by AdaBoost training.

do not present a consistent appearance. The AdaBoost algorithm cannot tolerate small size, low quality, intensity variation, or unclear boundaries in the license plate images, features that are frequently found in images of speeding vehicles on highways during a 24 hour period. For 692×516 pixel images, the OpenCV algorithm requires a computing time of 0.5 sec to detect the license plate.

5. CONCLUSIONS

We propose a dynamic programming-based approach to detect the four main numbers on the license plate. The proposed search method is very fast and can identify the plate numbers using an energy minimization model for the geometric configuration of successively located numeric characters. Most of the typical algorithms used to find the position of the license plate use color information and therefore fail to find the plate location when the body of the vehicle and its license plate have similar colors. Because the proposed method uses a gray-scale image, the color variation or environmental conditions have little effect on the extraction performance of the characters on the plate and consistent operation of the overall system is therefore possible under the severely varying lighting conditions experienced outdoors over a period of 24 hours.

ACKNOWLEDGEMENT–This research was financially supported by the Ministry of Education, Science Technology (MEST) and Korea Industrial Technology Foundation (KOTEF) through the Human Resource Training Project for Regional Innovation. This work is financially supported by the Ministry of Education and Human Resources Development (MOE), the Ministry of Commerce, Industry and Energy (MOCIE) and the Ministry of Labor (MOLAB) through the fostering project of the Industrial-Academic Cooperation Centered University.

REFERENCES

Chang, S. L., Chen, L. S., Chung, Y. C. and Chen, S. W. (2004). Automatic license plate recognition. *IEEE Trans. Intelligent Transportation Systems* **5**, **1**, 42−53.

Chen, X., Yuille, A. L. (2004). Detecting and reading text in natural scenes. *Proc. Int. Conf. Computer Vision and Pattern Recognition*, 366−373.

Civica Company, http://www.platescan.com

DataWorks Plus Company, http://www.digitalcrimescene.com

Dlagnekov, L. (2005). *Video-based Car Surveillance: License Plate, Make, and Model Recognition*. M. S. Thesis. University of California San Diego.

Duda, R. O., Hart, P. E. and Stock, D. G. (2001). *Pattern Classification*. Wiley-Interscience. New York.

Hontani, H. and Koga, T. (2001). Character Extraction Method without Prior Knowledge on Size and Position Information. *IEEE Int. Conf. Vehicle Electronics*, 67−72.

Intel Company, (2007). http://www.intel.com/ technology/computing/opencv/

Jung, H. G., Lee, Y. H., Kim, D. S., Yoon, P. J. and Kim, J. H. (2007). Stereo vision-based forward obstacle detection. *Int. J. Automotive Technology* **8**, **4**, 493−504.

Lee, M. W. and Nevatia, R. (2007). Body part detection for human pose estimation and tracking. *Proc. IEEE Workshop on Motion and Video Computing*, 23−30.

Li, Y., Suen, C. Y. and Cheriet, M. (2005). A threshold selection method based on multiscale and graylevel co-occurrence matrix analysis. *Proc. Int. Conf. Document Analysis and Recognition*, 575−579.

Neapolitan, R. and Naimipour, K. (1998). *Foundation of Algorithms Using C++ Pseudocode*. Jones and Bartlett. Toronto.

Nijhuis, J. A. G., Brugge, M. H. T., Helmholt, J. P. W., Pluim, L., Spaanenburg, R. S., Venema, R. S. and Westenberg, M. A. (2005). Car license plate recognition with neural networks and fuzzy logic. *IEEE Int. Conf. Neural Networks*, **5**, 2232−2236.

Nixon, M. and Aguado, A. (2002). *Feature Extraction and Image Processing*. Elsevier. 69−79.

Park, T. H. and Kim, N. (2007). A dynamic programming approach to PCB assembly optimization for surface mounters. *Int. J. Control, Automation, and Systems* **5**, **2**, 192−199.

Sauvola, J., Haapakoski, S., Kauniskangas, H., Seppanen, T. and Pietikainen, M. (1997). A distributed management system for testing document image analysis algorithms. *Int. Conf. Document Analysis and Recognition*, 989−995.

Viola, P. and Jones, M. J. (2004). Robust real-time face detection. *Int. J. Computer Vision* **57**, **2**, 137−154.

Wu, Y., Lian, F., Huang, C. and Chang, T. (2007). Image processing techniques for lane-related information extraction and multi-vehicle detection in intelligent highway vehicles. *Int. J. Automotive Technology* **8**, **4**, 513−520.

Zhang, H., Jia, W., He, X. and Wu, Q. (2006). Learning-based license plate detection using global and local features. *Proc. Int. Conf. Pattern Recognition*, 1102−1105.