Car License Plate Detection Method for Malaysian Plates-Styles by Using a Web Camera

Abbas M. Al-Ghaili*, Syamsiah Mashohor, Abdul Rahman Ramli and Alyani Ismail
Department of Computer and Communication Systems, Faculty of Engineering, Universiti Putra Malaysia, 43400 UPM, Serdang, Selangor, Malaysia
**E-mail: abbasghaili@yahoo.com*

ABSTRACT

Recently, license plate detection has been used in many applications especially in transportation systems. Many methods have been proposed in order to detect license plates, but most of them work under restricted conditions such as fixed illumination, stationary background, and high resolution images. License plate detection plays an important role in car license plate recognition systems because it affects the accuracy and processing time of the system. This work aims to build a Car License Plate Detection (CLPD) system at a lower cost of its hardware devices and with less complexity of algorithms' design, and then compare its performance with the local CAR Plate Extraction Technology (CARPET). As Malaysian plates have special design and they differ from other international plates, this work tries to compare two likely-design methods. The images are taken using a web camera for both the systems. One of the most important contributions in this paper is that the proposed CLPD method uses Vertical Edge Detection Algorithm (VEDA) to extract the vertical edges of plates. The proposed CLPD method can work to detect the region of car license plates. The method shows the total time of processing one 352x288 image is 47.7 ms, and it meets the requirement of real time processing. Under the experiment datasets, which were taken from real scenes, 579 out of 643 images were successfully detected. Meanwhile, the average accuracy of locating car license plate was 90%. In this work, a comparison between CARPET and the proposed CLPD method for the same tested images was done in terms of detection rate and efficiency. The results indicated that the detection rate was 92% and 84% for the CLPD method and CARPET, respectively. The results also showed that the CLPD method could work using dark images to detect license plates, whereas CARPET had failed to do so.

Keywords: Adaptive thresholding, Car License Plate Detection (CLPD), CAR Plate Extraction Technology (CARPET), Vertical Edge Detection Algorithm (VEDA)

INTRODUCTION

Car license plate recognition system is an image processing technology used to identify vehicles by capturing their Car License Plates (CLPs). The car license plate recognition technology is also known as automatic number-plate recognition, automatic vehicle identification, car license plate recognition or optical character recognition for cars. Car License Plate Detection and Recognition System (CLPDRS) became an important area of research due to its various applications such as in the payment of parking fee, highway toll fee, traffic data collection, and crime prevention (Huda, Khalid, Yosuf and Omar, 2007; Thanongsak and Kosin, 1999). Usually, a CLPDRS consists of three parts, namely license plate detection, character segmentation, and character recognition. Among these parts, license plate detection is the most important part in the system because it affects the accuracy of the system (Bai and Liu, 2004).

Received: 26 December 2008
Accepted: 3 December 2009
*Corresponding Author

There are many issues which should be resolved in order to create a successful and fast Car License Plate Detection System (CLPDS); these include poor image quality, plate sizes and designs, processing time, as well as background details and complexity. The need for car identification is increasing for many reasons such as crime prevention, vehicle access control, and border control. In order to identify a car, features such as its model, colour, format, and license plate number can be used (Fukumi, Takeuchi, Fukumoto, Mitsukura and Khalid, 2005; Lee, Kim and Kim, 1994; Parisi, Claudio, Lucarelli and Orlandi, 1998).

In vehicle tracking systems, cameras are used and installed in front of policemen cars to identify those vehicles. Usually, numerous vehicle tracking and pursue systems use outstanding cameras (Naito, Tsukada, Yamada, Kozuka and Yamamoto, 2000), and this leads to cost increment of the system involving both hardware and software. Many methods have been proposed in various Intelligent Transportation System (ITS) applications, but the CLPDRS is usually based on images acquired at a 640 x 480 resolution (Wu, Chen, Wu and Shen, 2006). Enhancing the performance of the CLPD method, such as reducing computation time and algorithm complexity, or even building of License Plate Recognition (LPR) system with lower cost of its hardware devices, will make it more practical and usable than before. This paper proposed a new Vertical Edge Detection Algorithm (VEDA) for detecting vertical edges. In addition, a VEDA-based method was also proposed for car license plate detection in which a web-camera (with 352 x 288 resolutions) was used instead of a more sophisticated one. In this work, the web-camera was used to capture the images and then an off-line process was performed to detect the plate detection from the whole scene image.

This paper is organized as follows: Section 2 includes a brief review of some related work and Section 3 describes the proposed method for CLPD. The experimental results and discussion are presented in Section 4 and the conclusion is given in Section 5.

RELATED WORK

A number of methods have been used for license plate region detection; these include morphological operations (Hsieh, Yu and Chen, 2002; Lensky, Jo and Gubarev, 2006; Martin, Garcia and Alba, 2002), edge extraction (Le, Seok and Lee, 2002; Parker and Federl, 1997; Yu and Kim, 2000; Zhang, Jia, He and Wu, 2006; Zheng, Zhao and Wang, 2005), combination of gradient features (S. Kim, Kim, Ryu and Kim, 2002), a neural network for colour classification (Lee *et al*., 1994), and vector quantization (Rovetta and Zunino, 1999).

Due to ambient lighting conditions, interference characters, and other problems, it is difficult to detect license plates in complex conditions. Some of the previous license plate detection methods are restricted to work under certain conditions such as fixed backgrounds (Bai, Zhu and Liu, 2003), and known colour (Debi, Chae and Jo, 2008; S. K. Kim, Kim and Kim, 1996; Ron and Erez, 2002; Shahaf, Timor and Erez, 2002).

In the previous years, some researchers have been working on license plate detection in complex conditions. Among other, Kim *et al*. (2002) proposed a license plate detection algorithm using both statistical features and license plate templates. After the statistical features were used to select the Regions of Interest (ROI), the license plate templates were applied to match the ROI. In many cases, general license plate templates are very difficult to be constructed. Moreover, their algorithm could work on a fixed scale. Hence, the application of this algorithm is restricted.

The work by Zheng *et al*. (2005) as well as Thanongsak and Kosin (1998: 1999) made use of image enhancement and Sobel operator to extract the vertical edges of the image of cars. They used an algorithm to remove most of the background and noisy edges. Finally, they searched the plate region using a rectangular window in the residual edge image. Recently, the authors (Abolghasemi and Ahmadyfard, 2007) improved the work by Zheng *et al*. (2005) by enhancing the low quality input image and then extracting the vertical edges. Then, they used morphological filtering to constitute some regions as plate regions.

Bai *et al*. (2003) proposed an algorithm for license plate detection to monitor highway ticketing systems. Their algorithm presented a linear filter in order to overcome the influence of lights. In addition, the vertical edge detection was also used in order to detect the license plate region. However, their algorithm could only work with fixed backgrounds.

Recently, a CAR Plate Extraction Technology (CARPET), which is used to detect Malaysian car license plate and recognize license numbers, was proposed by Ming (2004a). This method uses a stationary camera in car parking area for capturing the samples. Therefore, this method can process fixed background images.

While the method discussed above processes Malaysian plates, one of the objectives in this work is to compare the proposed CLPD method with this particular method in order to evaluate them. Low quality image produced by a web-camera is one of the problems and difficulties that should be solved. The usage of low quality image in CLPDS can reduce the size of the memory used while processing the input image (Naito *et al*., 2000). This advantage helps in building applicable hardware system at low cost and with smaller memory size. This paper proposed a new method for CLPDS in which a web-camera is used for acquiring images.

THE PROPOSED METHOD FOR THE CLPD

Overview

In this paper, the proposed CLPD method is introduced and discussed. Then, a comparison between CARPET and the proposed CLPD methods in this study is presented. The proposed CLPD method consists of some processes as follows: First, the colour input image is converted to grey-scale image, and then, adaptive thresholding is applied on the image to constitute the binarized image. After that, an enhancement of the binarized image is performed such as noise removal. Then, the vertical edges are extracted by using VEDA. The details of the plate are highlighted based on the colour information with help of the VEDA output form. Later, some statistical and logical operations are used to detect and search for the true candidate region. Finally, the true plate region is detected in the original image. *Fig. 1* shows the flowchart of the proposed CLPD method.

Adaptive Thresholding (AT)

After colour input image is converted to grey scale values (0-255), an adaptive thresholding process (Bradley and Roth, 2007; Shafait, Keysers and Breuel, 2007) is applied on the grey scale image input to constitute the binarized image. The researchers in (Bradley and Roth, 2007) have recently proposed real-time adaptive thresholding using mean of a local window, where the local mean is computed using an integral image.

First of all, in order to get a good adaptive threshold, the method proposed by Bradley and Roth (2007) is used. The authors proposed the real-time adaptive thresholding method using mean of a local window, where the local mean was computed using an integral image. The integral image is pre-computed for every pixel. The integral image is computed for every pixel g(i,j), as in Bradley and Roth (2007):

$$IntgrlImg(i,j) = \begin{cases} sum(i) & if(j=0) \\ IntgrlImag(i,j-1) + sum(i) & otherwise \end{cases} \quad (1)$$

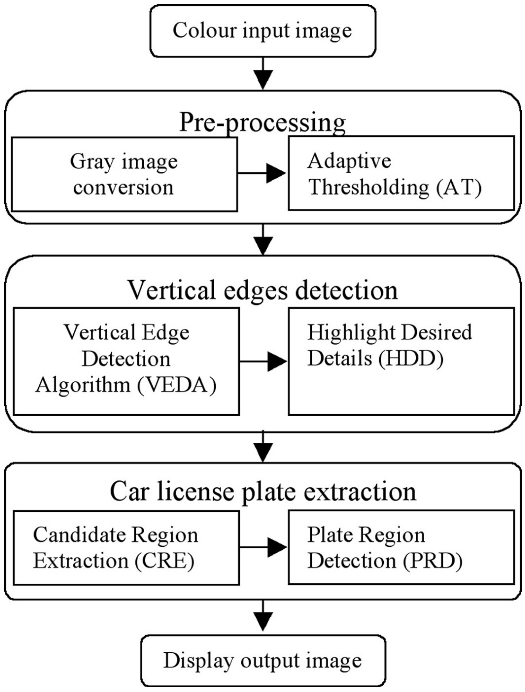

Fig. 1: Flowchart of the proposed method

Where *IntgrlImg(i,j)* represents the integral image for pixel *(i,j)* and *sum(i)* represents the summation of the pixel values for all the *i*-values with one column j^{th}, and these are computed according to Bradley and Roth (2007):

$$sum(i)\Big|_{j^{th}} = \sum_{x=0}^{i} g(x,y)\Big|_{y=j^{th}} \tag{2}$$

Whereas *g(x,y)* represents the input value.

By using (2) and (1), the integral image is created. The next step is to perform thresholding for each pixel. In order to get the adaptive threshold value for *(i, j)*, this criterion is tested for all the pixel values according to Bradley and Roth (2007):

$$o(i,j) = \begin{cases} 0 & g(i,j)*S^2(1-T)*sum \\ 255 & otherwise \end{cases} \tag{3}$$

Where *g(i,j)* represents the input image values, S^2 represents the *S*x*S* region, where $S = \frac{image\,width}{8}$, *T* is a constant and *T=0.15*, and *sum* represents the summation of the intensities of the grey values for the window size $(i+\frac{S}{2}, j+\frac{S}{2}), (i+\frac{S}{2}, j-\frac{S}{2}), (i-\frac{S}{2}, j+\frac{S}{2})\, and\, (i-\frac{S}{2}, j-\frac{S}{2})$. Once the integral image is obtained, the intensity summation for any window size can be computed using two subtraction and one additional operation instead of the summation over all the pixel values within that particular window (Porikli and Tuzel, 2006):

$$sum = \left(IntgrlImg(i+\frac{s}{2},j+\frac{s}{2})\right) - \left(IntgrlImg(i+\frac{s}{2},j-\frac{s}{2})\right) - \left(IntgrlImg(i-\frac{s}{2},j+\frac{s}{2})\right) + \left(IntgrlImg(i-\frac{s}{2},j-\frac{s}{2})\right) \quad (4)$$

Once the *sum* in (4) is calculated, the threshold is calculated and tested for each pixel value using Eq. (3).

Fig. 2(a) shows the input image and *Fig. 2(b)* illustrates the results after the application of AT.

(a) Input image *(b) Threshold image*

Fig. 2: Adaptive thresholding

Vertical Edge Detection Algorithm (VEDA)

There are two procedures involved in this step. First, thin lines are removed to reduce details of the image. As it is noticeable from the binarized image, some details are not important and contain very thin lines. Therefore, eliminating them could enhance the detection rate and reduce computation time. Second, the vertical lines are extracted.

For removing the lines which are not parts of license plate, the mask in *Fig. 3* is proposed and discussed in (Al-Ghaili *et al.,* 2008).

Each case of *a, b, c* and *d* represents two corresponding grey scale values each time the mask moves through the grey scale value in the threshold image *g(x,y)* values, where *x* and *y* represent the rows and columns locations, respectively.

The binarized image shown in *Fig. 2 (b)* is enhanced by applying the proposed mask (see *Fig. 3*) and the output is shown in *Fig. 4*.

In order to extract the vertical edges, a 2x4 mask was proposed in Al-Ghaili *et al*. (2008), as shown in *Fig. 5*. Basically, the proposed mask consists of 3 smaller masks, namely the left mask 2x1, centre mask 2x2, and right mask 2x1. *Fig. 5* shows the design of the proposed mask, where *(x,y), x,* and *y* represent the current processed pixel location at point (0,1) as the centre of the proposed mask, the rows or the height of the image, and the columns or the width of the image, respectively.

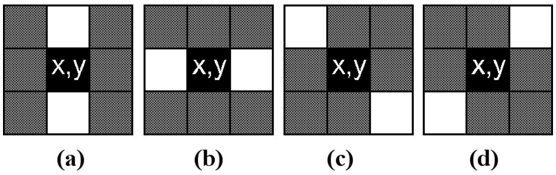

Fig. 3: The mask used for removing too long and thin lines

Fig. 4: The enhanced image after applying the proposed mask

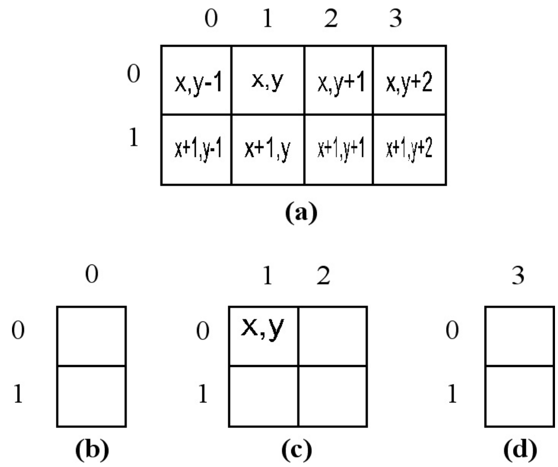

Fig. 5: The design of the proposed mask: (a) moving mask; (b) left mask (0,0), (1,0); (c) centre mask (0,1), (0,2), (1,1), (1,2); (d) right mask (0,3), (1,3)

The enhanced binarized image (*Fig. 4*) is scanned by moving the proposed mask *(Fig. 5* (*a*)) from left to right and from top to bottom. *Fig. 6* shows the output of this process.

Highlight Desired Details Based on VEDA (HDD)

After applying VEDA, the next step is to highlight the desired details such as plate details and vertical edges in the image. HDD performs NAND-AND operation for each two corresponding pixel values taken from both ULEA and VEDA output images. NAND-AND operation for this process is illustrated in *Fig. 7* as shown.

After all the pixels are scanned, the regions which probably contain plate details are then highlighted, as shown in *Fig. 8*.

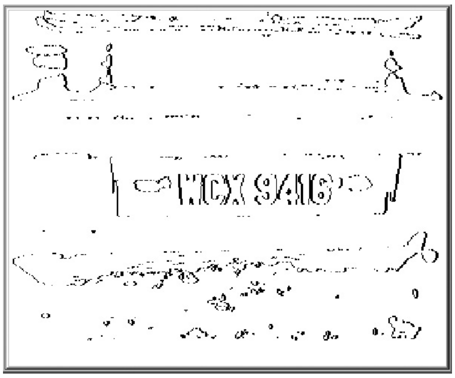

Fig. 6: VEDA output

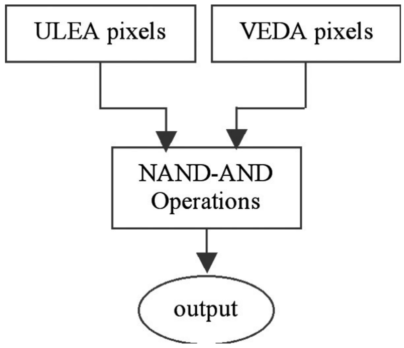

Fig. 7: HDD output generation by using NAND-AND operation

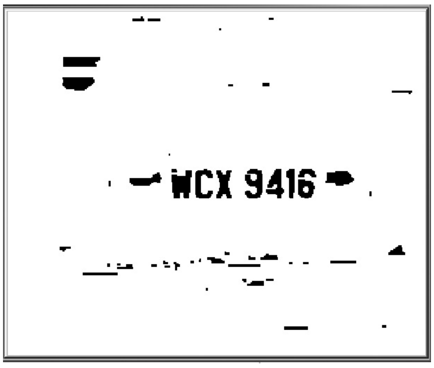

Fig. 8: HDD output

All the pixels in the vertical edge image (for example in *Fig. 6*) are scanned. When there are two neighbour black pixels, followed by one black pixel as in VEDA output form (see *Fig. 6*), the two following tests will be checked to highlight the desired details by drawing black horizontal lines connecting the two vertical edges. First, these two vertical edges should be surrounded by a black background, as shown in the threshold image (see *Fig. 4*). Second, the value of the horizontal distance (hd) should be in range of $2<hd<33$, where hd represents the length between the two vertical edges. Prior to this, the hd was computed from the tested images. The hd-length is appropriate to remove very short lines and long lines, as well as to retain details of the plate. This scanning process will be started from left to right and from top to bottom. After all the pixels have been scanned, the regions which probably contain plate details are shown in *Fig. 8*.

Candidate Regions Extraction (CRE)

In this process, the candidate regions are extracted. Some statistical operations are used in this step. First, the number of lines which have been extracted in the HDD process per each row are counted and stored in a matrix variable, *HwMnyLines[a], a=0, 1… height-1*. This process is then repeated for all the pixel values in the HDD image. Then, the image is divided into smaller groups using Equation (5).

$$how\_mny\_groups = \frac{height}{C} \tag{5}$$

In this equation, the total number of groups is equivalent to the total number of rows divided by C, where *how_mny_groups*, *height* and C represent the total number of groups, the total number of image rows, and group-constant, respectively. In the present work, C was chosen to represent one group (set of rows). For the methodology employed in this study, C=10 because each 10 rows could save the processing time and also keep the desired details clearly while processing the images. Probably, more or less than 10, either loses much desired details or wastes much time for processing the image. This particular step helps to distinguish the regions that may have plate details.

Fig. 9 shows an example of the total number of lines vs. each group (for the image shown in *Fig. 8*). In *Fig. 9*, *group* represents values (0-28). This range is calculated using (5), as follows:

$$how\_mny\_groups = \frac{288}{10} \approx 29$$

Where, the total number of image-rows equals to 288, and the total number of groups equals to 29 (0-28) groups. The dimensions of the processed input image are 352 x 288 pixels.

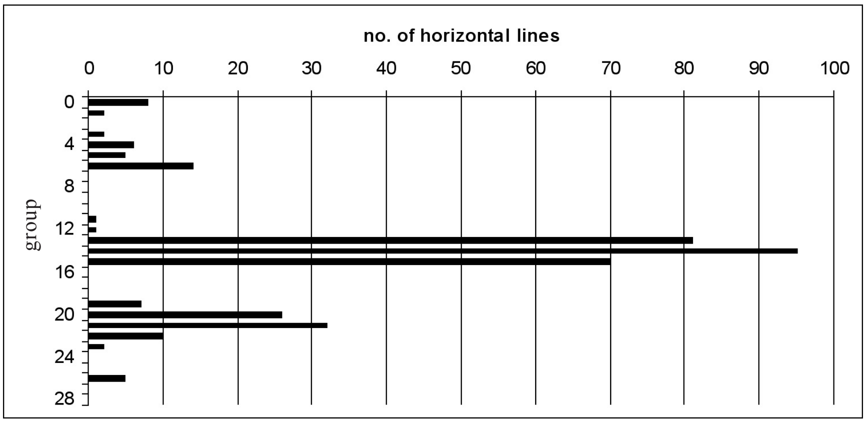

Fig. 9: Number of horizontal lines vs. group

This figure denotes that the highest values of lines frequency are located in the plate region. The second step in this process is performing a threshold in order to eliminate the groups which do not belong to the plate region and to keep others. The remaining groups after the thresholding step should have the plate details. Therefore, the locations of those groups are saved to be processed later by their indexes. While these groups may be one connected region or more than one region, each region is considered as a separate plate region. Thus, this step aims to extract both the upper and lower boundaries for each region that may contain plate region by their indexes. Finally, the horizontal boundaries above and below each candidate region will be drawn in the process. *Fig. 10* shows the result of drawing boundaries of the candidate regions in the input image.

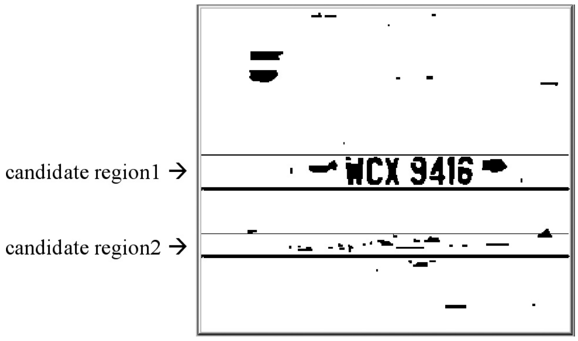

Fig. 10: Output of drawing candidate regions boundaries

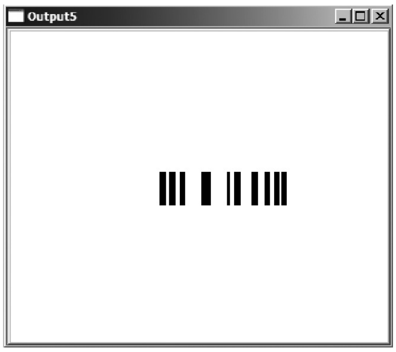

Fig. 11: Draw lines for some columns in candidate regions

Plate Region Selection (PRS)

This process aims to select one region as a plate region and extract the plate region, using the following steps: First, check the blackness-ratio in each candidate region column. Check the ratio of blackness in each column of candidate regions. If the ratio exceeds 50% of the height of the candidate region columns, consider the current checked column as a part of the plate. Then, draw a new black vertical line and replace the current column with the new one. Otherwise, replace the current column with the white background (see *Fig. 11*). *Fig. 11* shows only the vertical lines in which their blackness ratios exceed 50%.

Fig. 12 shows the blackness frequency for both the candidate regions as illustrated in *Fig. 10*. The true candidate region has the blackness frequency more than the others.

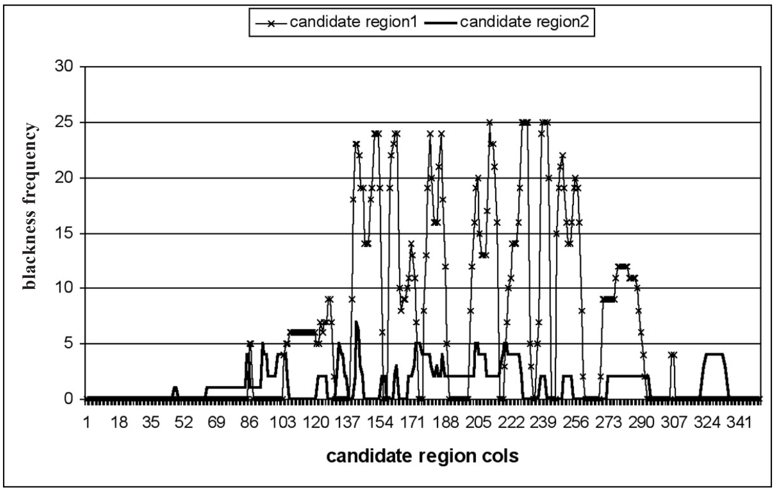

Fig. 12: Both candidate region columns vs. blackness frequency

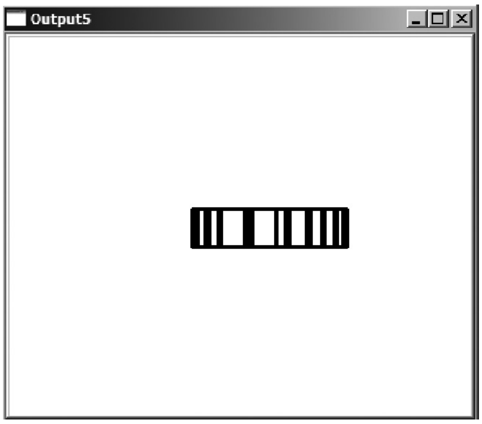

Fig. 13: Extract plate area

Fig. 14: Plate area detection

Next, make a vote for all the candidate regions. As the background colour of the plate is black, the top and down neighbours of the selected columns in the previous step will be checked. If there is enough ratio of blackness, the current checked region will get one vote. The process will be done for all the columns in all the candidate regions and all these candidate regions are compared to select one suitable candidate. Hence, the candidate region with the highest vote values is the selected region as the true license plate. Finally, by tracking the black vertical lines, the plate area can then be detected and extracted as well. The region with the highest value of voting will be detected, while the black vertical lines will be checked and tracked. The region will be detected as the plate area, as shown in *Fig. 13*. *Fig. 14* shows the result derived for the license plate detection after the whole procedure has been performed.

PRS factor (K-factor)

Sometimes, when the images are blurry or the plate region is defected, the ratio value should be dynamically changed. Therefore, *K-factor* is proposed to overcome the problem and this is discussed in this section.

As mentioned above, each column of the candidate regions is checked one by one. If the column blackness ratio exceeds 50%, the current column belongs to the license plate region and this particular column (see *Fig. 10*) can then be replaced by a vertical black line in the resulting image (see *Fig. 11*). Hence, each column is checked using this condition:

If *temp* is greater than or equals to *0.5*cnst*, the current column is then considered as a part of the license plate region, where *temp*, *cnst* represent the total number of the black pixels per each column in the current candidate region and the height of the column of the candidate region, respectively. This condition with a fixed value (0.5) is used with non-blurry images. Unfortunately, some candidate regions will be neglected by the software in case the ratio of blackness to the total length (height) of the candidate region is less than 50%. Therefore, the condition is changed to be less than 50% according to the ratio of the blurry level or the degradation, and then *If (temp≥K*cnst)*, where *K* represents the blurry level factor. The value of *K* must be reduced when the blurry level or degradation is high in order to highlight more important details; on the contrary, *K* must be increased when the blurry level is less.

Fig. 15: K-value (K≥0.5)

Fig. 16: K-value (K≥0.4)

When *K*-value is high (*K*≥0.5), the drawn black columns are extracted and shown from non-blurry image. In this case, if the processed image is blurry, the black pixels will be reduced and the candidate regions will be neglected to be detected and the plate region will be missed or inefficiently detected, as shown in *Fig. 15*.

Using the same example (*K*-value≥0.4), more required details will be shown, as illustrated in *Fig. 16*.

It is clear that the result presented in *Fig. 16* is better than the one in *Fig. 15*. However, it still needs a little bit of enhancement to improve it. *Fig. 17* shows the result for detection when *K*-value≥0.3.

By comparing the three results obtained above, it can be stated that the best result is in *Fig. 17*, i.e. when the *K*-value is ≥0.3.

Therefore, *K* is proportional reverse to blurry level or degradation, and thus:

$$K \propto \frac{1}{blurrylevel\ ratio}$$

By controlling the *K* value, some candidate regions will be processed and taken into account. Thus, this procedure will enhance both the detection rate and the whole performance of the CLPD method, as demonstrated in the experimental results.

Fig. 17: K-value (K≥0.3)

EXPERIMENTAL RESULTS AND DISCUSSION

This part is divided into two sections; first, an evaluation of the CLPD method is introduced, and second, a comparison between the CLPD and CARPET is discussed.

Evaluation of the Proposed CLPD Method

There were 643 images of Malaysian license plates captured from parking cars using a web camera. The resolution is 352x288. The images were captured at day time and in various weather conditions. These samples were standardized in this experiment. These datasets were selected based on gathering and capturing Malaysian plates from UPM campus.

This experiment had three datasets. There were 100 images in Dataset 1. In this test, 85 images were successfully detected, while 15 images were failed to be detected. The percentage of successful detection was 85%.

Meanwhile, there were also 100 samples for Dataset 2. In this test, 90 images were successfully detected, but it failed to detect 10 other images. The percentage of successful detection was 90%.

For Dataset 3, 404 out of 443 images were successfully detected. The percentage for the car license plate detection accuracy was 91.2%.

From this evaluation, 579 out of 643 images were successfully detected, and thus, the car license plate detection accuracy was 90%.

The datasets were taken using a web camera for many Malaysian car plates which were from UPM Campus. The detection rate in Dataset 1 before PRS-factor was proposed. Meanwhile, the detection rate for Dataset 2 and Dataset 3 was after. Therefore, the PRS-factor was found to have enhanced the whole accuracy of the proposed CLPD method.

Fig. 18 shows the increase in the percentage of detection rate for all the datasets before and after the PRS-factor.

Based on the data presented in *Fig. 18*, the successful detection rate was increased from 87% to 91%. *Fig. 19* shows the average processing time for 100 test images. As illustrated in *Fig. 19*, most of the processed images required 47 ms to be completed, while some images with complex background took more processing time in general because details from their complex background were recorded more and calculated from 100 experimental images. The average processing time for one 352x288 image was 47.7 ms, and this fulfilled the real time processing requirement.

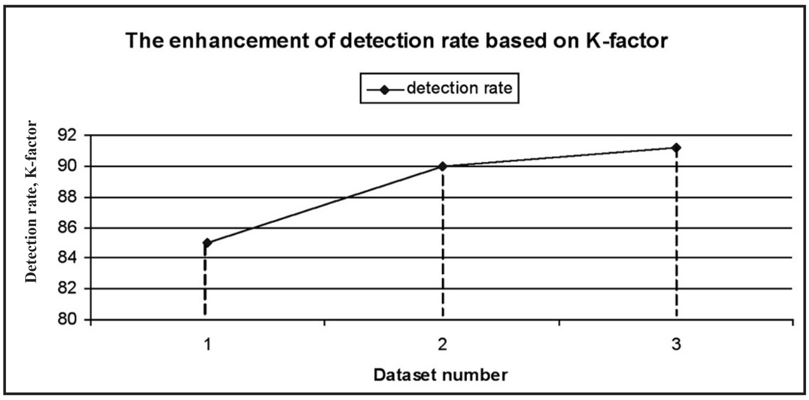

Fig. 18: The increase in the detection rate before and after PRS-factor

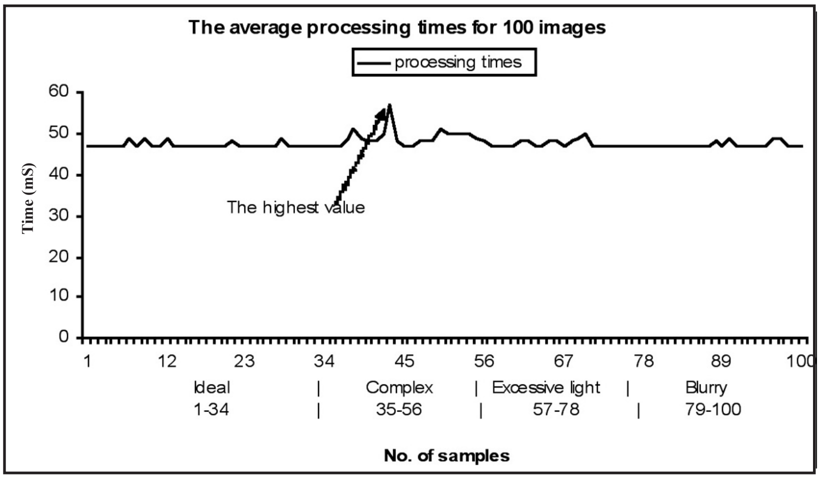

Fig. 19: The average processing time for 100 images

There are seven stages involved in a complete processing of an image in the proposed CLPD method. The average processing time (in msec) for the seven stages of the proposed method is listed in Table 1. A lot of the time is consumed during the second stage, i.e. AT.

TABLE 1
The average processing time for the seven stages involved in the proposed method

Conversion	AT	ULEA	VEDA	HDD	CRE	PRS & PRD	Total time (ms)
1.1	15	5.2	7	7.9	3.3	8.2	47.7

CARPET vs. the Proposed CLPD Method

This section introduces a comparison between CARPET and the proposed CLPD method. As there are a number of images available in the demo version of the CARPET software [(1)], twenty five images from both the datasets (CLPD & CARPET) were randomly taken and tested using the CARPET software (Ming, 2004b), and then tested by the proposed CLPD in this work.

Detection rate

Table 2 shows the experimental results for the test carried out in this study. From the following table, the proposed method was found to outperform CARPET in terms of detection rate.

TABLE 2
Comparison of detection rates between CARPET and the proposed CLPD method

Methods	Plates detected	Plates not detected	Detection rates (%)
CARPET	21	4	84
The proposed CLPD	23	2	92

In this test, CARPET fails to detect the plate efficiently when the colour of car body is dark. The two missing plates (see Table 2) for the proposed CLPD method were because of the blurry level and the similarity in the colours of the body and plate of the car.

Efficiency

As shown by the data presented in Table 3, CARPET was found to have failed to detect the license plate in four samples, while the proposed CLPD method in two samples. These were samples are 3, 13, 22, and 24. In samples 3 and 13, both the CARPET and CLPD methods had failed to detect the license plates. In samples 22 and 24, the CLPD method detected the license plates correctly whereas CARPET had failed to do so. Samples 3 and 13 were blurred and had similarity in their body-and-plate colours, respectively. Samples 22 and 24 were dark images.

From Table 3, the proposed CLPD method is shown to be insusceptible for changes in image condition such as darkness (as noticeable in samples 22 and 24), while CARPET is found to be sensitive in this case. Therefore, the proposed CLPD method can be used for dark images and it is more efficient than CARPET in such case.

(1) Integrated Testing Version 1.0, © 2006. see: “http://www.projekcarpet.com/download/demo.zip”

TABLE 3
CARPET vs. CLPD in different conditions

Method / Sample no.	CARPET	CLPD
1	y	y
2	y	y
3	n	n
4	y	y
5	y	y
6	y	y
7	y	y
8	y	y
9	y	y
10	y	y
11	y	y
12	y	y
13	n	n
14	y	y
15	y	y
16	y	y
17	y	y
18	y	y
19	y	y
20	y	y
21	y	y
22	n	y
23	y	y
24	n	y
25	Y	y

CONCLUSIONS

A vertical-edges car license plate detection method has been proposed and discussed in this paper. The vertical edges were detected based on VEDA. The proposed CLPD method was compared with the CARPET system for some tested images. This paper confirmed that the proposed CLPD method has higher detection rate and works more efficiency in different image conditions. From the experimental results, 579 out of 643 images were successfully detected. The CLPD method has a detection rate of 90% for the whole tested images. Using a total of five images, the comparison between CARPET and the proposed CLPD method revealed that the percentage of the detection rate was 92% (the proposed CLPD) and 84% (CARPET), respectively. This paper also showed that the detection rate using the PRS-factor could enhance the accuracy and overall performance of the proposed CLPD method.

ACKNOWLEDGEMENT

This work was supported by the Ministry of Higher Learning, Malaysia, under the Fundamental Research Grant no. 5523427.

REFERENCES

Abolghasemi, V. and Ahmadyfard, A. (2007). Improved image enhancement method for license plate detection. Paper presented at the *Proceedings of the 15th International Conference on Digital Signal Processing* (DSP), Iran.

Al-Ghaili, A. M., Mashohor, S., Ismail, A. and Ramli, A. R. (2008). A new vertical edge detection algorithm and its application. Paper presented at the *IEEE International Conference on Computer Engineering & Systems* (ICCES 2008), Cairo, Egypt.

Bai, H. and Liu, C. (2004). A hybrid license plate extraction method based on edge statistics and morphology. Paper presented at the *Proceedings of the 17th International Conference on Pattern Recognition*, UK.

Bai, H., Zhu, J. and Liu, C. (2003). A fast license plate extraction method on complex background. Paper presented at the *Proceedings of the IEEE International Conference on Intelligent Transportation Systems*, China.

Bradley, D. and Roth, G. (2007). Adaptive Thresholding using the Integral Image. *Journal of Graphics Tools*, *12*(2), 13–21.

Debi, K., Chae, H.-U. and Jo, K.-H. (2008). Parallelogram and histogram based vehicle license plate detection. Paper presented at the *IEEE International Conference on Smart Manufacturing Application*, Korea.

Fukumi, M., Takeuchi, Y., Fukumoto, H., Mitsukura, Y. and Khalid, M. (2005). Neural network based threshold determination for Malaysia license plate character recognition. Paper presented at the *Proceedings of the 9th International Conference on Mechatronics Technology*, Malaysia.

Hsieh, J.-W., Yu, S.-H. and Chen, Y.-S. (2002). Morphology-based license plate detection from complex scenes. Paper presented at the *Proceedings of the 16th International Conference on Pattern Recognition*, Canada.

Huda, S. N., Khalid, M., Yosuf, R. and Omar, K. (2007). Comparison of feature extractors in license plate recognition. Paper presented at the *Proceedings of the IEEE 1st Asia International Conference on Modelling & Simulation* (AMS'07), Thailand.

Kim, S., Kim, D., Ryu, Y. and Kim, G. (2002). A robust license-plate extraction method under complex image conditions. Paper presented at the *Proceedings of 16th International Conference on Pattern Recognition*, Canada.

Kim, S. K., Kim, D. W. and Kim, H. J. (1996). A recognition of vehicle license plate using a genetic algorithm based segmentation. Paper presented at the *Proceedings of International Conference on Image Processing*, Switzerland.

Le, S.-H., Seok, Y.-S. and Lee, E.-J. (2002). Multi-national integrated car-license plate recognition system using geometrical feature and hybrid pattern vector. Paper presented at the *Proceedings of the International Technical Conference on Circuits/ Systems, Computers and Communication*, Thailand.

Lee, E. R., Kim, P. K. and Kim, H. J. (1994). Automatic recognition of a car license plate using color image processing. Paper presented at the *Proceedings of the IEEE International Conference on Image Processing*, USA.

Lensky, A. A., Jo, K.-H. and Gubarev, V. V. (2006). Vehicle license plate detection using local fractal dimension and morphological analysis. Paper presented at the *Proceedings IEEE 1st International Forum Strategic Technology*, South Korea.

Martin, F., Garcia, M. and Alba, J. L. (2002). New methods for automatic reading of VLP's (Vehicle License Plates). Paper presented at the *Proceedings of IASTED International Conference on Signal Processing, Pattern Recognition, and Applications*, Greece.

Ming, C. C. *et al.* (2004a). ProjekCARPET, car license plate extraction & recognition technology. Retrieved on 2008 from http://www.projekcarpet.com/aboutus.asp.

Ming, C. C. *et al.* (2004b). ProjekCARPET, car license plate extraction & recognition technology (Demo Version). Retrieved on 2008 from http://www.projekcarpet.com/download/demo.zip.

Naito, T., Tsukada, T., Yamada, K., Kozuka, K. and Yamamoto, S. (2000). Robust license-plate recognition method for passing vehicles under outside environment. *IEEE Transactions on Vehicular Technology, 49*(6), 2309-2319.

Parisi, R., Claudio, E. D., Lucarelli, G. and Orlandi, G. (1998). Car plate recognition by neural networks and image processing. Paper presented at the *Proceedings of the IEEE International Symposium on Circuits and Systems*, USA.

Parker, J. R. and Federl, P. (1997). An approach to license plate recognition. Paper presented at the *Proceedings of Visual Interface,* Canada.

Porikli, F. and Tuzel, O. (2006). Fast construction of covariance matrices for arbitrary size image windows. Paper presented at the *Proceedings of the International Conference Image Processing*, USA.

Ron, B.-H. and Erez, J. (2002). A real-time vehicle license plate recognition (LPR) system. Retrieved on 2008 from http://visl.technion.ac.il/projects/2003w24/.

Rovetta, S. and Zunino, R. (1999). License-plate localization by using vector quantization. Paper presented at the *Proceedings of the International Conference on Acoustics, Speech and Signal Processing,* USA.

Shafait, F., Keysers, D. and Breuel, T. M. (2007). Efficient implementation of local adaptive thresholding techniques using integral images. Paper presented at the *Document Recognition and Retrieval XV*.

Shahaf, O., Timor, A. and Erez, J. (2002). Real time license plate recognition in a video movie. Retrieved on 2008 from http://visl.technion.ac.il/projects/2002w03/.

Thanongsak, S. and Kosin, C. (1998). Extracting of car license plate using motor vehicle regulation and character pattern recognition. Paper presented at the *Proceedings of the 1998 IEEE Asia-Pacific Conference on Circuit and Systems*.

Thanongsak, S. and Kosin, C. (1999). The recognition of car license plate for automatic parking system. Paper presented at the *Proceedings of the 5th International Symposium on Signal Processing and Its Applications*, Australia.

Wu, H.-H. P., Chen, H.-H., Wu, R.-J. and Shen, D.-F. (2006). License plate extraction in low resolution video. Paper presented at the *Proceedings of the IEEE 18th International Conference on Pattern Recognition*, Hong Kong.

Yu, M. and Kim, Y. D. (2000). An approach to Korean license plate recognition based on vertical edge matching. Paper presented at the *Proceedings of the IEEE Plate Detection in Various Conditions* and the *Proceedings of the International Conference on Systems, Man, and Cybernetics*, USA.

Zhang, H., Jia, W., He, X. and Wu, Q. (2006). A fast algorithm for license. *IEEE International Conference on Systems, Man, and Cybernetics.* Taiwan.

Zheng, D., Zhao, Y. and Wang, J. (2005). An efficient method of license plate location. *Pattern Recognition Letters, 26*, 2431-2438.