Vertical-Edge-Based Car-License-Plate Detection Method

Abbas M. Al-Ghaili, Syamsiah Mashohor, Abdul Rahman Ramli, and Alyani Ismail

***Abstract*—This paper proposes a fast method for car-license-plate detection (CLPD) and presents three main contributions. The first contribution is that we propose a fast vertical edge detection algorithm (VEDA) based on the contrast between the grayscale values, which enhances the speed of the CLPD method. After binarizing the input image using adaptive thresholding (AT), an unwanted-line elimination algorithm (ULEA) is proposed to enhance the image, and then, the VEDA is applied. The second contribution is that our proposed CLPD method processes very-low-resolution images taken by a web camera. After the vertical edges have been detected by the VEDA, the desired plate details based on color information are highlighted. Then, the candidate region based on statistical and logical operations will be extracted. Finally, an LP is detected. The third contribution is that we compare the VEDA to the Sobel operator in terms of accuracy, algorithm complexity, and processing time. The results show accurate edge detection performance and faster processing than Sobel by five to nine times. In terms of complexity, a big-O-notation module is used and the following result is obtained: The VEDA has less complexity by K^2 times, whereas K^2 represents the mask size of Sobel. Results show that the computation time of the CLPD method is 47.7 ms, which meets the real-time requirements.**

***Index Terms*—Adaptive thresholding (AT), car-license-plate detection (CLPD), Sobel operator, vertical edge detection algorithm (VEDA).**

I. Introduction

THE car-license-plate (CLP) recognition system is an image processing technology used to identify vehicles by capturing their CLPs. The CLP recognition technology is known as automatic number-plate recognition, automatic vehicle identification, CLP recognition, or optical character recognition for cars. The CLP detection and recognition system (CLPDRS) became an important area of research due to its various applications, such as the payment of parking fees, highway toll fees, traffic data collection, and crime prevention [1], [2].

Usually, a CLPDRS consists of three parts: license-plate (LP) detection (LPD), character segmentation, and character recognition. Among these, LPD is the most important part in the system because it affects the system's accuracy [3].

Manuscript received January 25, 2012; revised May 8, 2012, July 9, 2012, and August 9, 2012; accepted September 18, 2012. Date of publication October 3, 2012; date of current version January 14, 2013. This work was supported by the Higher Learning Education Ministry of Malaysia under Fundamental Research Grant 5523427. The review of this paper was coordinated by Dr. W. Zhuang.

The authors are with the Faculty of Engineering, Universiti Putra Malaysia, 43400 Serdang, Malaysia (e-mail: abbasghaili@yahoo.com; syamsiah@eng.upm.edu.my; arr@eng.upm.edu.my; alyani@eng.upm.edu.my).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2012.2222454

There are many issues that should be resolved to create a successful and fast CLP detection system (CLPDS), e.g., poor image quality, plate sizes and designs, processing time, and background details and complexity. The need for car identification is increasing for many reasons such as crime prevention, vehicle access control, and border control. To identify a car, features such as model, color, and LP number can be used [4]–[6].

In vehicle tracking systems, cameras are used and installed in front of police cars to identify those vehicles. Usually, numerous vehicle tracking and pursue systems use outstanding cameras [7], and this leads to cost increment of the system in both hardware and software. Since many methods have been proposed in various intelligent transportation system applications, the CLPDRS is usually based on an image acquired at 640 × 480 resolution [8]. An enhancement of CLPD method performance such as reduction of computation time and algorithm complexity, or even the build of the LP recognition system with lower cost of its hardware devices, will make it more practical and usable than before. This paper proposed a method for CLPD, in which a web camera with 352 × 288 resolution is used instead of a more sophisticated web camera. In this paper, the web camera is used to capture the images, and an offline process is performed to detect the plate from the whole scene image.

Vertical edge extraction and detection is an important step in the CLPDRS because it affects the system's accuracy and computation time. Hence, a new vertical edge detection algorithm (VEDA) is proposed here to reduce the computation time of the whole CLPD method [9]. This paper is organized as follows. Section II introduces a brief of related work. Section III describes two parts. The first part discusses in detail our proposed approach to vertical edge detection, i.e., using an unwanted-line elimination algorithm (ULEA) and the VEDA. The second part discusses the proposed CLPD method. Experimental results and discussion are presented in Section IV. Section V draws our conclusions.

II. Related Work

There are several LPD methods that have been used before, such as morphological operations [10]–[12], edge extraction [13]–[18], combination of gradient features [19], salience features [20], a neural network for color [5] or grayscale [21] classification, and vector quantization [22].

Due to ambient lighting conditions, interference characters, and other problems, it is difficult to detect LPs in complex conditions. Some of previous LPD methods are restricted to work under certain conditions, such as fixed backgrounds [23] and known color [24]–[27].

In previous years, some researchers have been working on LPD in complex conditions. Kim *et al.* [19] proposed an LPD algorithm using both statistical features and LP templates. After the statistical features were used to select the regions of interest (ROIs), LP templates were applied to match the ROI. In many cases, general LP templates are very difficult to be constructed. Moreover, their algorithm can work on a fixed scale. Hence, the application of this algorithm is restricted. In [28], Matas and Zimmermann proposed an algorithm to detect LPs under various conditions. Their algorithm used character regions as basic units of LPs, which make the algorithm quite robust to viewpoint and illumination. However, this algorithm could hardly highlight characters overlapping from the true LPs. In [29], an LPD algorithm using color edge and fuzzy disciplines has been proposed. However, it can only be used with certain colors.

Vertical edge extraction is one of the most crucial steps in CLPD because it influences the whole system to correctly detect the LP [30]. An edge map has vastly reduced complexity, and it retains the important structure present in the original image [31]. Thus, a vertical edge map has been used for LPD for many years [3], [15], [23], [25]. The given algorithms used a one-directional Sobel operator to extract the vertical edges. Nevertheless, some undesired details such as horizontal edges are kept in such vertical edge map. Therefore, these details can increase the processing time and reduce the system accuracy. In [2], [17], [32], an image enhancement and Sobel operator was used to extract the vertical edges of the car image. They used an algorithm to remove most of the background and noisy edges. Finally, they searched the plate region by a rectangular window in the residual edge image. Recently, Abolghasemi and Ahmadyfard [33] have improved the method proposed in [17] 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. Zhang *et al.* [16] defined a new vertical gradient map to extract statistical features. The authors constructed two cascade classifiers based on statistical and Haar features to decrease the complexity of the system and to improve the detection rate. However, this method will take much processing time even with low-quality images.

Bai *et al.* [23] proposed an algorithm for LPD for monitoring the highway ticketing systems. Their algorithm presented a linear filter to smooth the image and to overcome the influence of light. In addition, vertical edge detection was used for suppressing horizontal noise. Then, the edge density was measured and compared with the true plate region density. A nonlinear filter was applied to remove the narrow horizontal lines. Finally, a connected component analysis algorithm was applied to show and to locate the LP features. However, their algorithm works better with a fixed background and a stationary camera.

The most common and earliest edge detection algorithms are those based on the gradient, such as the Sobel operator [34] and the Roberts operator [35]. Numerous previous methods have used the Sobel operator to extract the vertical edges in CLPDSs [16], [17], [23], [32], [36]. In this paper, we proposed the VEDA to extract vertical edges.

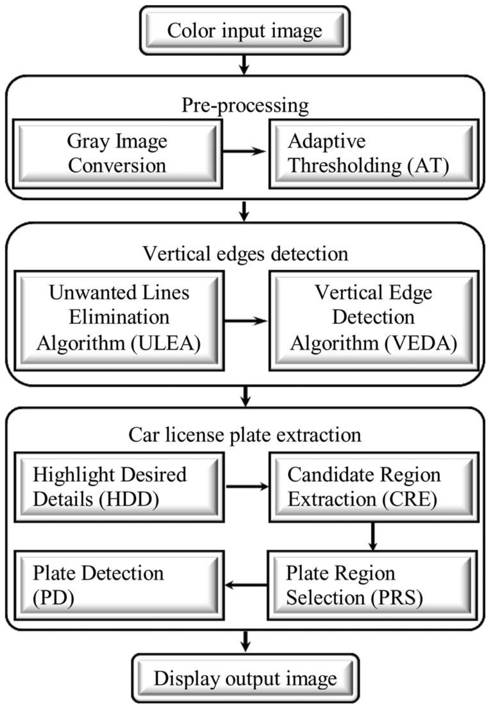
Color input image
Pre-processing
Gray Image Conversion
Adaptive Thresholding (AT)
Vertical edges detection
Unwanted Lines Elimination Algorithm (ULEA)
Vertical Edge Detection Algorithm (VEDA)
Car license plate extraction
Highlight Desired Details (HDD)
Candidate Region Extraction (CRE)
Plate Detection (PD)
Plate Region Selection (PRS)
Display output image

Fig. 1. Flowchart of the proposed method.

A low-resolution image produced by a web camera is one of the issues that should be solved. The purpose of using low-resolution images with the CLPDS is to gain three advantages: less memory size, less cost, and low computing time. This paper proposed a new method for CLPD, in which a web camera is used to capture input images.

III. Proposed Method for Car License Plate Detection

A. Overview

This paper has three contributions: The VEDA is proposed and used for detecting vertical edges; the proposed CLPD method processes low-quality images produced by a web camera, which has a resolution of 352×288 with 30 fps; and the computation time of the CLPD method is less than several methods. In this paper, the color input image is converted to a grayscale image, and then, adaptive thresholding (AT) is applied on the image to constitute the binarized image. After that, the ULEA is applied to remove noise and to enhance the binarized image. Next, the vertical edges are extracted by using the VEDA. The next process is to detect the LP; the plate details are highlighted based on the pixel value with the help of the VEDA output. Then, some statistical and logical operations are used to detect candidate regions and to search for the true candidate region. Finally, the true plate region is detected in the original image. The flowchart of the proposed CLPD method is shown in Fig. 1.

B. AT

After the color input image is converted to grayscale, an AT process [37], [38] is applied to constitute the binarized image.

Bradley and Roth [37] recently proposed real-time AT using the mean of a local window, where local mean is computed using an integral image. To get a good adaptive threshold, the method proposed in [37] is used.

1) Technique of AT: The AT technique used in this paper is just a simple extension of Bradley and Roth's [37] and Wellner's methods [39]. The idea in Wellner's algorithm is that the pixel is compared with an average of neighboring pixels. Specifically, an approximate moving average of the last S pixels seen is calculated while traversing the image. If the value of the current pixel is T percent lower than the average, then it is set to black; otherwise, it is set to white. This technique is useful because comparing a pixel to the average of neighboring pixels will keep hard contrast lines and ignore soft gradient changes. The advantage of this technique is that only a single pass through the image is required. Wellner uses one eighth of the image width for the value of S and 0.15 for the value of T to yield the best results for a variety of images. The value of T might be a little bit modified from the proposed value by Wellner depending on the used images; whereas it should be in the range $0.1 < T < 0.2$ in our method.

However, Wellner's algorithm depends on the scanning order of pixels. Since the neighborhood samples are not evenly distributed in all directions, the moving average process is not suitable to give a good representation for the neighboring pixels. Therefore, using the integral image in [37] has solved this problem.

2) AT Formulations: The first step is to compute the integral image. Initially, the summation of the pixel values for every column jth through all row values i will be computed using

$$\text{sum}(i)\Big|_{j^{\text{th}}} = \sum_{x=0}^{i} g(x,y)\Big|_{y=j^{\text{th}}} \tag{1}$$

where $g(x,y)$ represents the input values, and $\text{sum}(i)|_{j^{\text{th}}}$ represents all cumulative gray values of $g(x,y)$ for the column jth through all rows of image $I = 0, 1, \ldots$ height.

Then, the integral image can then be computed for every pixel as in (2):

$$\text{IntgrlImg}(i,j) = \begin{cases} \text{sum}(i), & \text{if } j = 0 \\ \text{IntgrlImg}(i,j-1) + \text{sum}(i), & \text{otherwise} \end{cases} \tag{2}$$

where $\text{IntgrlImg}(i,j)$ represents the integral image for pixel (i,j).

The next step is to perform thresholding for each pixel. To do so, first, the intensity summation for each local window should be computed by using two subtraction operations and one addition operation [40] as follows:

$$\begin{aligned} \text{sum}_{\text{window}} = &\left(\text{IntgrlImg}\left(i+\frac{s}{2}, j+\frac{s}{2}\right)\right) \\ &- \left(\text{IntgrlImg}\left(i+\frac{s}{2}, j-\frac{s}{2}\right)\right) \\ &- \left(\text{IntgrlImg}\left(i-\frac{s}{2}, j+\frac{s}{2}\right)\right) \\ &+ \left(\text{IntgrlImg}\left(i-\frac{s}{2}, j-\frac{s}{2}\right)\right) \end{aligned} \tag{3}$$

where $\text{sum}_{\text{window}}$ represents the summation of the intensities of the gray values for a specified local window, in which the currently binarized pixel is centering in. The boundaries of the window can be represented by

$$\left(i+\frac{s}{2}, j+\frac{s}{2}\right), \left(i+\frac{s}{2}, j-\frac{s}{2}\right), \left(i-\frac{s}{2}, j+\frac{s}{2}\right), \left(i-\frac{s}{2}, j-\frac{s}{2}\right)$$

and s represents the local window size/lengths for the computed IntgrlImg, whereas $s = \text{image width}/8$.

Therefore, to compute the adaptive threshold value for the image, in which $g(i,j) \in [0, 255]$ is the intensity of the pixel located at (i,j), threshold $t(i,j)$ for each pixel has to be computed first as follows:

$$t(i,j) = (1-T) \times \text{sum}_{\text{window}} \tag{4}$$

where $t(i,j)$ represents the threshold for each pixel at location (i,j), and T is a constant, i.e., $T = 0.15$. This value is the optimal value for best thresholding performance for the whole images after testing on many images and is assessed visually.

The criterion in the following is applied on every pixel to output the threshold value of that pixel:

$$o(i,j) = \begin{cases} 0, & g(i,j) \times S^2 < t(i,j) \\ 255, & \text{otherwise} \end{cases} \tag{5}$$

where $o(i,j)$ represents the adaptive threshold output value of pixel $g(i,j)$, and S^2 represents the computed area of the local window for the selected region.

3) Effect of T Value on AT: Here, the effect of the T value will be explained to evaluate the sensitivity of the image. Two examples will be given here: The first example has a lesser value of T than the pregiven value, and the other example has a higher value. In (4), if $T = 0.05 \rightarrow t(i,j) = 0.95 \times \text{sum}_{\text{window}}$, then condition $g(i,j) \times S^2 < t(i,j)$ in (5) will be true in most cases. Therefore, adaptive threshold $o(i,j) = 0$ and a lot of gray regions $g(i,j)$ in the input image will appear in the binarized image as black regions. On the contrary, if $T = 0.29 \rightarrow t(i,j) = 0.71 \times \text{sum}_{\text{window}}$, then the condition in (5) will be false. That is, important details will disappear. After testing a lot of values, the most adequate value for T has been found as 0.15 for all of the input images.

As a result, the decrement of T below 0.15 will affect in constituting new black regions, and *vice versa*, the increment of T above 0.15 will affect in eliminating important details.

Fig. 2(a) shows the input image, and the result is shown in Fig. 2(b) after AT is applied.

C. ULEA

Thresholding process in general produces many thin lines that do not belong to the LP region. In Fig. 2(b), we can see that there are many long foreground lines and short random noise edges beside the LP region. These background and noise edges are unwanted lines. These lines may interfere in the LP location. Therefore, we have proposed an algorithm to eliminate them from the image. This step can be considered as a morphological

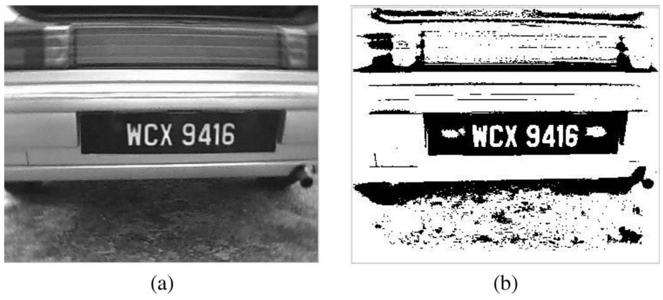

Fig. 2. Image binarization. (a) Input image. (b) Thresholded image.

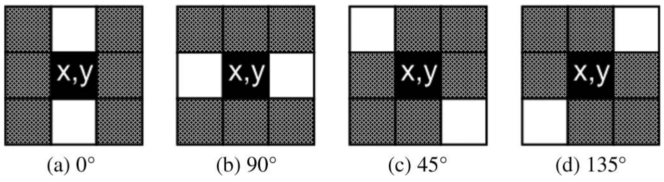

Fig. 3. Four cases for converting the center pixel to background. (a) Horizontal. (b) Vertical. (c) Right inclined. (d) Left inclined.

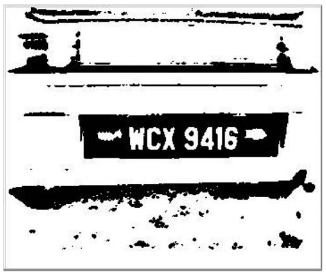

Fig. 4. ULEA output.

operation and enhancement process. There are four cases in which unwanted lines can be formed. In the first case, the line is horizontal with an angle equal to 0° as (−). In the second case, the line is vertical with an angle equal to 90° as (|). In the third case, the line is inclined with an angle equal to 45° as (/). In the fourth case, the line is inclined with an angle equal to 135° as (\). Therefore, the ULEA has been proposed to eliminate these lines. In this step, while processing a binary image, the black pixel values are the background, and the white pixel values are the foreground. A 3 × 3 mask is used throughout all image pixels. Only black pixel values in the thresholded image are tested. To retain the small details of the LP, only the lines whose widths equal to 1 pixel are checked. Suppose that $b(x, y)$ are the values for thresholded image. Once, the current pixel value located at the mask center is black, the eight-neighbor pixel values are tested. If two corresponding values are white together, then the current pixel is converted to a white value as a foreground pixel value (i.e., white pixel).

Fig. 3 shows the possible cases in which the current pixel is converted to foreground pixel. Each case of (a)–(d) represents two corresponding values at each time that the mask moves through the image. Fig. 4 shows the output after the ULEA is performed, whereby many unwanted lines are removed from the image. This kind of image is nearly ready for a better segmentation process.

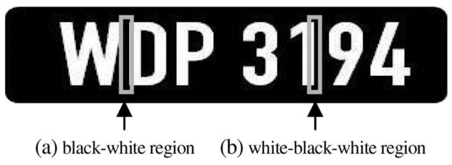

Fig. 5. Intersection of black–white and white–black regions.

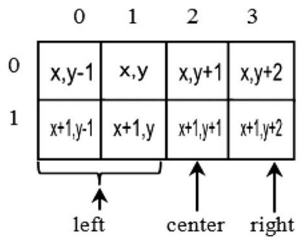

Fig. 6. Design of the proposed mask.

D. VEDA

The advantage of the VEDA is to distinguish the plate-detail region, particularly the beginning and the end of each character. Therefore, the plate details will be easily detected, and the character recognition process will be done faster. After thresholding and ULEA processes, the image will only have black and white regions, and the VEDA is processing these regions. The idea of the VEDA concentrates on intersections of black–white [see Fig. 5(a)] and white–black [see Fig. 5(b)]. A 2 × 4 mask is proposed for this process, as shown in Fig. 6, where x and y represent rows and columns of the image, respectively. The center pixel of the mask is located at points (0, 1) and (1, 1). By moving the mask from left to right, the black–white regions will be found. Therefore, the last two black pixels will only be kept. Similarly, the first black pixel in the case of white–black regions will be kept.

The proposed mask has the size of 2 × 4 to fulfill the following two criteria.

1) In this type of a mask, it is divided into three submasks: The first submask is the left mask "2 × 2," the second submask is the center "2 × 1," and the third submask is the right mask "2 × 1," as marked in Fig. 6. Simply, after each two pixels are checked at once, the first submask is applied so that a 2 pixel width "because two column are processed" can be considered for detecting. This process is specified to detect the vertical edges at the intersections of black–white regions. Similarly, the third submask is applied on the intersections of white–black regions. Thus, the detected vertical edge has the property of a 1 pixel width.
2) The number "2" points out the number of rows that are checked at once. The consumed time in this case can be less twice in case each row is individually checked.

To select the column at locations (0, 1) and (1, 1) to be the center of the proposed mask, two pixels and one pixel in the case of black–white and white–black regions are retained, respectively.

This process is performed for both of the edges at the left and right sides of the object-of-interest. The first edge can have a black-pixel width of 2, and the second edge can have a

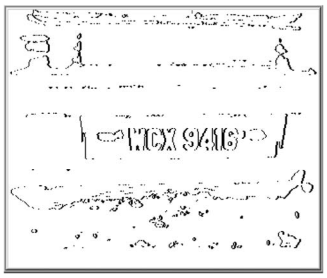

Fig. 7. VEDA output.

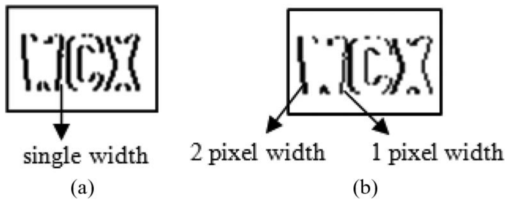

Fig. 8. Sobel and VEDA vertical edges' forms. (a) Sobel and (b) VEDA outputs.

black-pixel width of 1. The 2×4 mask starts moving from top to bottom and from left to right. If the four pixels at locations (0, 1), (0, 2), (1, 1), and (1, 2) are black, then the other mask values are tested if whether they are black or not. If the whole values are black, then the two locations at (0, 1) and (1, 1) will be converted to white. Otherwise, if column 1 and any other column have different values, the pixel value of column 1 will then be taken. This process is repeated with the whole pixels in the image. After applying this process on all of the pixels of the image in Fig. 4, the result is shown in Fig. 7.

1) Sobel's Output Form Versus VEDA Output Form: In Fig. 8, the output forms of Sobel and the VEDA are compared. For Sobel, both vertical edges of a detected object have the same thickness, as shown in Fig. 8(a). For the VEDA, there are two- and one-pixel thicknesses for each detected object, as shown in Fig. 8(b).

As VEDA's output form, the searching process for the LP details could be faster and easier because this process searches for the availability of a 2 pixel width followed by a 1 pixel width to stand for a vertical edge. This mechanism of searching could save more processing time. In addition, there is no need to search again once the 1 pixel width is faced. These two features could make the searching process faster and easier. A detailed explanation will be provided in the next process [highlighted desired details (HDDs)].

2) Sobel's Time Complexity Versus VEDA's Time Complexity: By using the big-O-notation module, the code complexity of Sobel and that of the VEDA in terms of time are compared for evaluation.

For the Sobel operator, the algorithm complexity is

$$Os(N) = O(N) \times O(M) \times O(K1) \times O(K2) \tag{6}$$

For the VEDA, the algorithm complexity is

$$Ov(N) = O(N) \times O(M) \tag{7}$$

where $Os(N)$ and $Ov(N)$ represent the code complexity of the Sobel algorithm and the VEDA, respectively, and $O(N)$ and O(M) represent the code's complexity of the first and second loops inside the Sobel and VEDA codes, respectively.

For the Sobel code, $O(K1) \times O(K2)$ represents the code complexity of the third and fourth loops as deduced from the Sobel code.

In (6), $K1$ and $K2$ represent the horizontal and vertical mask scales, respectively. Since these two masks are equal (i.e., 3×3), therefore

$$K1 = K2 = K. \tag{8}$$

Thus, the new formulation of (6) can be modified as

$$Os(N) = O(N) \times O(M) \times O(K^2). \tag{9}$$

Both (7) and (9) confirm that the VEDA has less complexity than the Sobel operator by K^2 times.

3) Sobel Versus Canny Versus VEDA: The Sobel operator was the most popular edge detection operator until the development of edge detection techniques with a theoretical basis. It proved popular because it overall gave a better performance than the other contemporaneous edge detection operators. The Canny edge detector has more steps than Sobel. As is known, Canny performs additional processing, i.e., the nonmaximum suppression that eliminates possible wide ridges that can result from the Canny enhancer or Sobel. Furthermore, where Sobel does simple thresholding, Canny combines the thresholding with contour following to reduce the probability of false contours [41]. Thus, Canny's computation time can be more than Sobel's computation time. We can conclude that Canny is perhaps the most popular edge detection technique at present [41] in terms of the accuracy but costs time. On the contrary, Sobel's accuracy is still accepted in vertical-edge-based applications. Therefore, the complexity of the Canny operator can be found by using big-O-notation, as follows:

$$Oc(N) = O(N) \times O(M) \times O(C^2) + 1. \tag{10}$$

By comparing (7)–(10), the complexity of VEDA is less than Canny.

E. HDDs Based on VEDA

After applying the VEDA, the next step is to highlight the desired details such as plate details and vertical edges in the image. The HDD performs NAND–AND operation for each two corresponding pixel values taken from both ULEA and VEDA output images. The NAND–AND procedure for this process is shown in Fig. 9.

This process depends on the VEDA output in highlighting the plate region. All the pixels in the vertical edge image (see Fig. 7) will be scanned. When there are two neighbor black pixels and followed by one black pixel, as in VEDA output form, the two edges will be checked to highlight the desired details by drawing black horizontal lines connecting each two vertical edges. First, these two vertical edges should be surrounded by a black background, as in the ULEA image in Fig. 4. Second, the value of horizontal distance hd represents

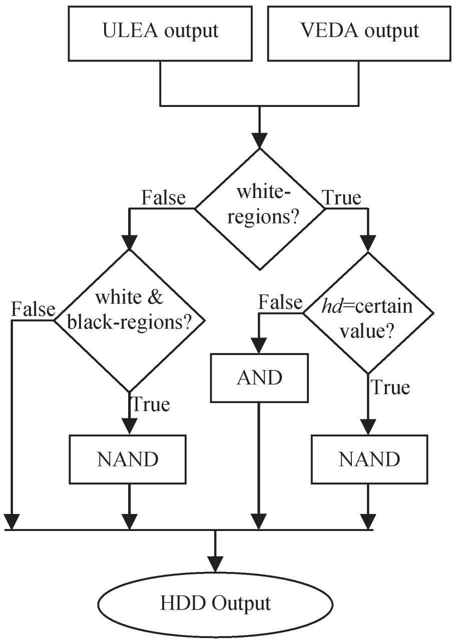

Fig. 9. NAND–AND procedure.

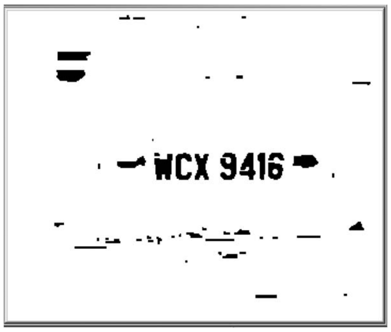

Fig. 10. HDD output.

the length between the two vertical edges of a single object. The hd has been computed using the test images. The hd value is selected to be suitable for removing long foreground and random noise edges that have not been eliminated earlier. This scanning process will start moving from left to right and from top to bottom. After all pixels are scanned, the regions in which the correct LP exists are highlighted, as shown in Fig. 10.

F. CRE

This process is divided into four steps as follows.

1) Count the Drawn Lines per Each Row: The number of lines that have been drawn per each row will be counted and stored in matrix variable HwMnyLines$[a]$, where $a = 0, 1, \ldots, \text{height-1}$.

2) Divide the Image into Multigroups: The huge number of rows will delay the processing time in the next steps. Thus, to reduce the consumed time, gathering many rows as a group is used here. Therefore, dividing the image into multigroups could be done using the following:

$$\text{how\_mny\_groups} = \frac{\text{height}}{C} \tag{11}$$

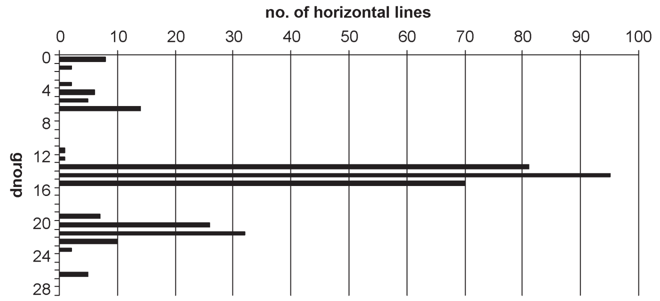

Fig. 11. Number of horizontal lines versus group.

where how_mny_groups represents the total number of groups, height represents the total number of image rows, and C represents the candidate region extration (CRE) constant. In this paper, C is chosen to represent one group (set of rows). For our methodology, $C = 10$ because each ten rows could save the computation time. In addition, it could avoid either loosing much desired details or consuming more computation time to process the image.

Therefore, each group consists of ten rows. Due to the HwMnyLines[a] values, some rows have a number of drawn horizontal lines, and this makes some groups to have horizontal lines. The step here is to store the total number of horizontal lines for each group. In this step, a matrix is created to store the total number of drawn lines for each ten rows (group). This step will distinguish the regions that might have LP details.

Fig. 11 shows an example for the total number of lines versus each group for the image shown in Fig. 10.

In Fig. 11, [group] represents values from 0 to 28. This range is calculated using (11) as

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

where the total number of image rows is equal to 288, and the total number of groups is equal to 29 groups.

This figure denotes that the highest values of line frequency are located in the LP region.

3) Count and Store Satisfied Group Indexes and Boundaries: Most of the group lines are not parts of the plate details. Therefore, it is useful to use a threshold to eliminate those unsatisfied groups and to keep the satisfied groups in which the LP details exist in. Each group will be checked; if it has at least 15 lines, then it is considered as a part of the LP region. Thus, the total number of groups including the parts of LP regions will be counted and stored. The remaining groups after thresholding step should have the LP details. Therefore, their locations are stored. The final step here is to extract both upper and lower boundaries of each satisfied group by using its own index.

This threshold (i.e., 15 lines) value is determined to make sure that the small-sized LP is included for a plate searching process. If the predefined threshold is selected with a less value than that given, wrong result can be yielded because noise and/or nonplate regions will be considered as parts of the true LP, even if they are not. Based on that, the optimal threshold for the best detection rate has been found is $\geq 1/20 \times \text{image\_height}$. Therefore, the threshold in our case is $\geq 1/20 \times 288 \geq 14.4 \equiv 15$.

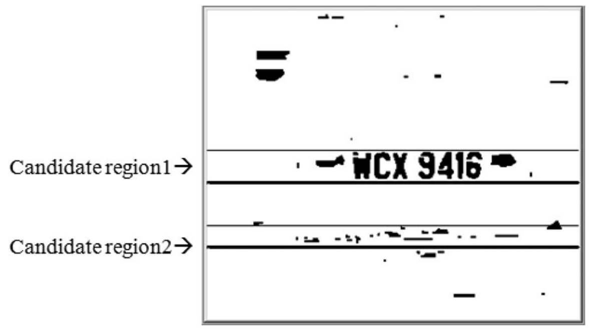

Fig. 12. Output of the boundaries of drawing candidate regions.

4) Select Boundaries of Candidate Regions: This step draws the horizontal boundaries above and below each candidate region. Fig. 12 shows the result of drawing candidate regions boundaries in the input image. As shown, there are two candidate regions interpreted from horizontal-line plotting, and these conditions require an additional step before the LP region can be correctly extracted.

G. PRS

This process aims to select and extract one correct LP. The process is discussed in five parts. The first part explains the selection process of the LP region from the mathematical perspective only. The second part applies the proposed equation on the image. The third part gives the proof of the proposed equation using statistical calculations and graphs. The fourth part explains the voting step. The final part introduces the procedure of detecting the LP using the proposed equation. The flowchart of plate region selection (PRS) and plate detection (PD) is also provided, as shown in Fig. 13.

1) Selection Process of the LP Region: As known, some of the processed images are blurry, or the plate region might be defected. The plate region can be checked pixel by pixel, whether it belongs to the LP region or not. A mathematical formulation is proposed for this purpose, and once this formulation is applied on each pixel, the probability of the pixel being an element of the LP can be decided.

As aforementioned, for the candidate regions, each column will be checked one by one. If the column blackness ratio exceeds 50%, then the current column belongs to the LP region; thus, this column (see Fig. 12) will be replaced by a vertical black line in the result image, as shown in Fig. 14. Hence, each column is checked by the condition that, if blckPix $\geq$ 0.5 $\times$ colmnHght, then the current column is an element of the LP region. Here, the blckPix represents the total number of black pixels per each column in the current candidate region, and the colmnHght represents the column height of the of the candidate region. This condition with a fixed value (0.5) is used with nonblurry images. However, some pixels of the candidate regions will not be detected in case the ratio of blackness to the total length (height) of the candidate region is greater than 50%. Therefore, the condition is changed to be less than 50%, according to the ratio of the blurry level or the deformation of the LP. The condition will be modified as follows. blckPix $\geq$ P_{RS} $\times$ colmnHght, where P_{RS} represents the PRS factor. The

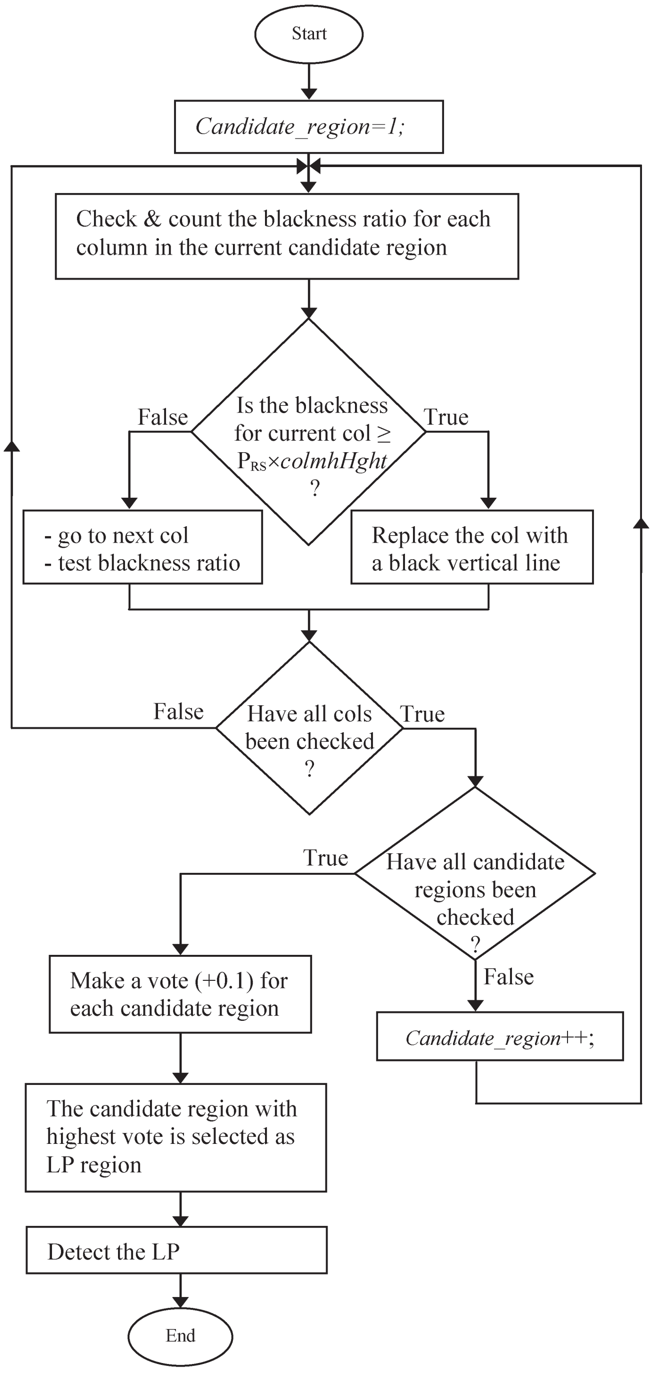
Start
Candidate_region=1;
Check & count the blackness ratio for each column in the current candidate region
Is the blackness for current col ≥ PRS×colmhHght ?
False
True
- go to next col
- test blackness ratio
Replace the col with a black vertical line
Have all cols been checked ?
False
True
Have all candidate regions been checked ?
True
False
Make a vote (+0.1) for each candidate region
Candidate_region++;
The candidate region with highest vote is selected as LP region
Detect the LP
End

Fig. 13. Flowchart of PRS and PD.

P_{RS} value is reduced when the blurry level is high to highlight more important details, and it is increased when the blurry level is less. Therefore, the mathematical representation for selecting the LP region can be formulated as follows:

$$C_{\text{region}} = \begin{cases} 0, & \text{blckPix} \geq P_{\mathrm{RS}} \times \text{colmnHght} \\ 255, & \text{otherwise} \end{cases} \tag{12}$$

where C_{region} represents the output value for the current pixel of the currently processed candidate region. If $C_{\text{region}} = 0$, consider the checked pixel as an element of the LP region; otherwise, consider it as background. The value of P_{RS} is

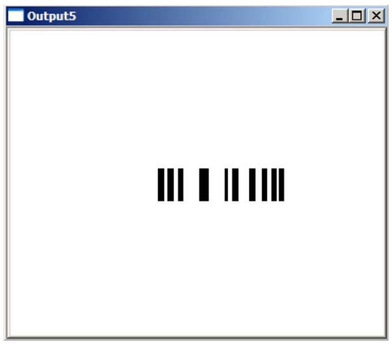

Fig. 14. Selection of the LP region.

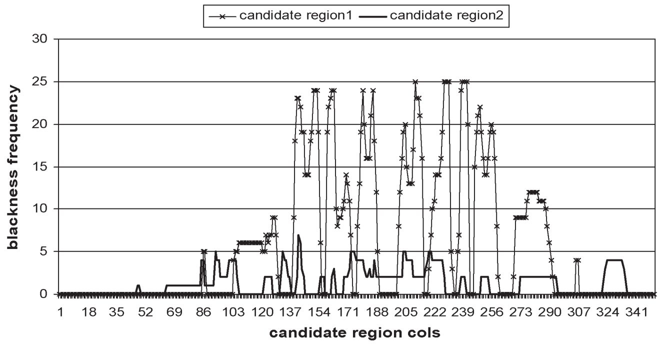

Fig. 15. Columns of both candidate regions versus blackness frequency.

automatically determined. Below is the proposed pseudocode in which (12) is applied.

```
**If** (blckPix $>= 0.5 \times$ colmhHght)
//i.e., $P_{\mathrm{RS}} = 50\%$; the blckPix $>= 0.5\times$ the column height of the correct region
**Then**
    $P_{\mathrm{RS}}$ is determined, and $= 0.5 \rightarrow$ the current column is a part of the LP;
**Else If** (blckPix $>= 0.4 \times$ colmhHght)
//i.e., $P_{\mathrm{RS}} = 40\%$; the blckPix $>= 0.4\times$ the column height of the correct region
**Then**
    $P_{\mathrm{RS}}$ is determined, and $= 0.4 \rightarrow$ the current column is a part of the LP;
**Else If** (blckPix $>= 0.3 \times$ colmhHght)
//i.e., $P_{\mathrm{RS}} = 30\%$; the blckPix $>= 0.3\times$ the column height of the correct region
**Then**
    $P_{\mathrm{RS}}$ is determined, and $= 0.3 \rightarrow$ the current column is a part of the LP;
**Else**
{
    The current column is not a part of the LP, and it belongs to background; Go to next column;
}
**End If**
```

2) Applying the Mathematical Formulation: After applying (12) on the image that contains the candidate regions shown in Fig. 12, the output can be shown in Fig. 14.

3) Proof of (12) Using Statistics: Fig. 15 shows the blackness frequency for both candidate regions shown in Fig. 12. From the plot, it can be observed that the true candidate region has higher blackness frequency than the wrong candidate.

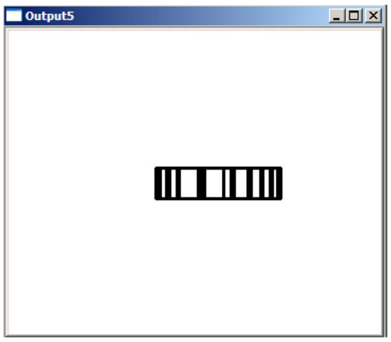

Fig. 16. Plate area detection.

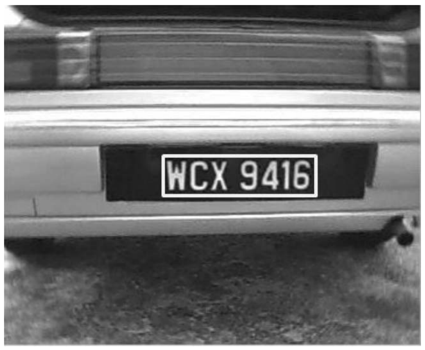

Fig. 17. LPD and LP extraction.

4) Making a Vote: The columns whose top and bottom neighbors have high ratios of blackness details are given one vote. This process is done for all candidate regions. Hence, the candidate region that has the highest vote values will be the selected region as the true LP. By tracking the black vertical lines, the plate area will be detected and extracted. In Fig. 16, the boundaries of the plate area are detected.

After the whole procedure has been performed, the final result for LPD is shown in Fig. 17.

5) LPD Using (12): An example is given to show how the value of the P_{RS} factor can whether increase or decrease the detection rate of the CLPD method. After applying (12) on any image obtained from the previous process (i.e., CRE), the output image contains a single candidate region. This effect of changing the P_{RS} value is given in the example. In Fig. 18(a), the boundaries of the plate area are detected, and the corresponding LP is marked, as shown in Fig. 18(b), whereas the P_{RS} value in this case has been set to be greater than or equal to 0.5. In Fig. 18(c) and (d), the P_{RS} value is set to be greater than or equal to 0.4. In the last case, the P_{RS} value is set to be greater than or equal to 0.3, as shown in Fig. 18(e) and (f).

By comparing the three obtained results, we can say that the best result is shown in Fig. 18(e) and (f). Thus, by controlling the P_{RS} value based on the processed image, the performance of the CLPD method is enhanced.

IV. Experimental Setup

A. Camera Setting

The prototype of the CLPD method is depicted in Fig. 19.

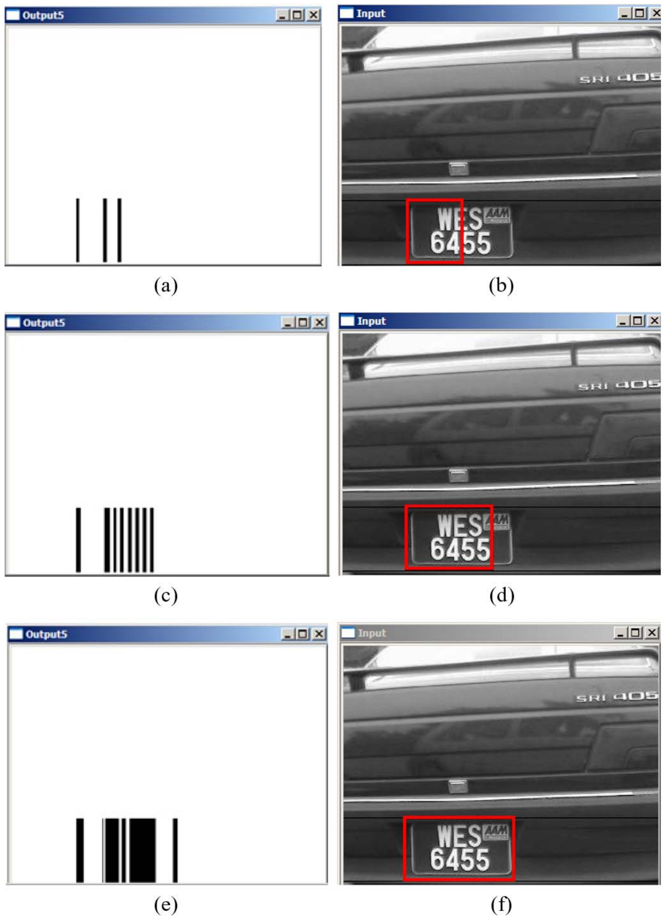

Fig. 18. LPD by modifying the $P_{\rm RS}$ value. (a) Plate area selection. (b) PD: $P_{\rm RS} \geq 0.5$. (c) Plate area selection. (d) PD: $P_{\rm RS} \geq 0.4$. (e) Plate area selection. (f) PD: $P_{\rm RS} \geq 0.3$.

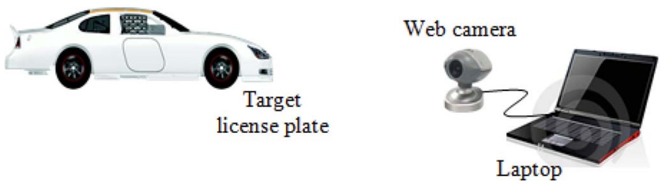

Fig. 19. Prototype of the CLPD method.

B. Implementation Steps

The implementation steps can be summarized as follows.

1) The web camera is set to active profile "live" and is connected to a laptop.
2) The devices described in (step 1) are taken to an outdoor environment.
3) The web camera is focused on the car LP.
4) The distance between the web camera and the LP ranges from 2 to 4 m.
5) The web-camera pan angles are in between $+20^\circ$ and -20°, whereas camera tilt is set from 0° to 20°. This procedure is shown in Fig. 20.
6) The captured samples contain different backgrounds and objects such as trees and two LPs.

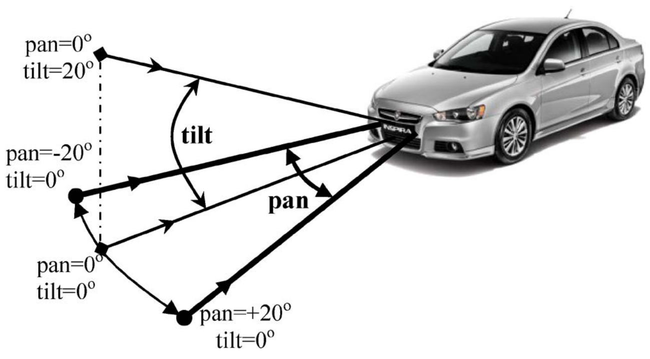

Fig. 20. Web-camera pan/tilt placement.

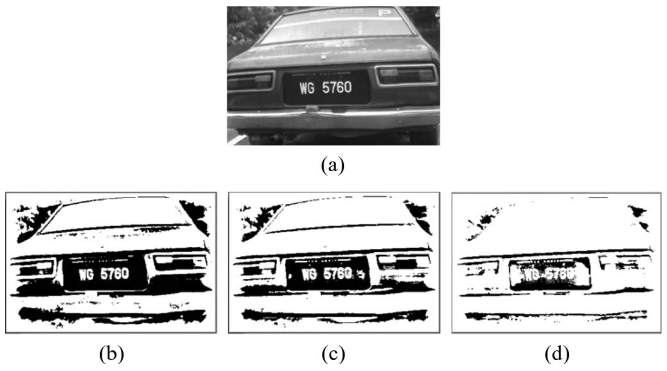

Fig. 21. Several binarized images with different values of T. (a) Original image. (b) Binarized image: $T = 0.04$. (c) Binarized image: $T = 0.15$. (d) Binarized image: $T = 0.03$.

V. Experimental Results and Discussion

Here, the AT process will be evaluated first. Then, the accuracy and the computation time of the VEDA are compared with that of the Sobel operator. Finally, the performance of the proposed CLPD method is evaluated. To carry out this evaluation and analysis, The VEDA and the Sobel algorithm are separately used to extract vertical edges. In the first evaluation, the CLPD method has been built as follows: ULEA→VEDA→HDD→CRE→PRS→PD. In the second evaluation, the CLPD method has been built as follows: ULEA→Sobel→HDD→CRE→PRS→PD. Our proposed method has been tested on the laptop with the following specifications: Core 2 CPUs with 1.83 GHz and 1 GB of RAM. The program is running under Windows XP SP2 written in C++ language with an OpenCV library.

A. AT Evaluation

To evaluate the effect of T mentioned in (4) and (5) on the performance of the CLPD method, the AT has been applied on the image shown in Fig. 21(a), with different values of T. The results are shown in Fig. 21(b)–(d).

As shown in Fig. 21(b), when T is equal to 0.04, there are adjacent regions around the LP. In Fig. 21(d), some important details such as the LP characters are partially eliminated and changed to background when the T is 0.3. Thus, the detection rate of the CLPD method might be affected in such a case.

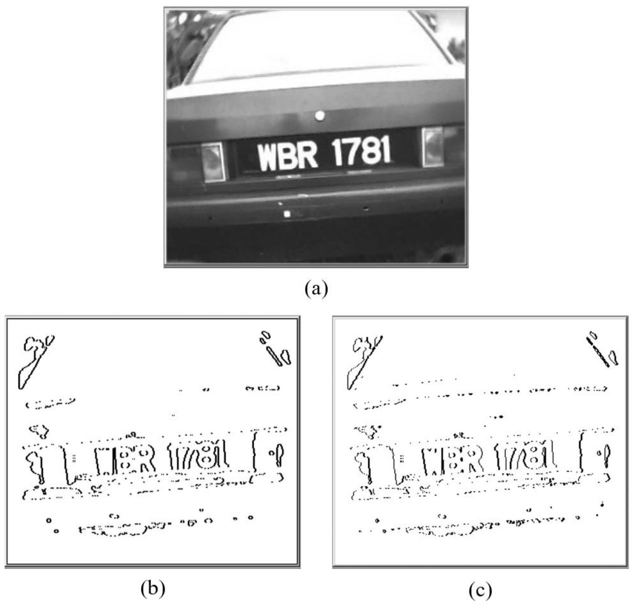

Fig. 22. Vertical edge detection using Sobel and the VEDA. (a) Input image. (b) Sobel's result ($time = 61$ ms). (c) VEDA result ($time = 7$ ms).

TABLE I
COMPARISON BETWEEN SOBEL AND VEDA

Image size / Method	Time (ms)		
	352×288	640×480	726×544
Sobel	61	133	164
VEDA	7	23	32
VEDA: Sobel speed	≈ 9 times	≈ 6 times	≈ 5 times

B. VEDA Versus Sobel

1) Accuracy: Both Sobel and VEDA have been running on same software and hardware environments to compare the accuracy, computation time, and complexity.

The accuracy and the computation time of the VEDA and Sobel are shown in Fig. 22. As stated, the VEDA consumed only 11% of Sobel's time, yet the appearance of edges is comparable.

2) Computation Time: Table I shows the computation time of the VEDA and Sobel for different image sizes. The best performance of VEDA is one ninth the computational times compared with Sobel. The ratio is getting less when the resolution of image increases.

3) Big-O-Notation-Module-Based Complexity Comparison: As discussed earlier in the proposed CLPD method, the VEDA has less complexity than the Sobel operator by K^2 times. Therefore, the VEDA is easier and a very simple tool to be used for extracting vertical edges.

C. Proposed CLPD Method Evaluation

Here, the samples collection will be explained. Then, the CLPD method will be evaluated using the VEDA and Sobel for extracting vertical edges. The proposed CLPD method will be compared with the Malaysian CAR Plate Extraction Technology (CARPET) [42] and several methods in terms of the computation time and the detection rate.

1) Sample Collection: The whole samples used in our experiments were captured in different parking areas and under various weather conditions such as rainy, sunny, and shady days taken from 8:00 A.M. to 6:00 P.M. They contain different backgrounds and objects such as trees and two LPs. A lot of difficulties in the experiments are faced such as blurriness, illumination changes, similarity between LP and car-body colors, LP sizes, LP designs, and double-row LPs, as shown in Fig. 23. Table II gives the experimental conditions. If the proposed algorithm is applied on a higher specification camera, the camera distance will also be increased.

2) CLPD Method Evaluation Using the VEDA: The whole tested images are 664, and they are classified into three groups based on the images quality, background complexity, and lighting effects. The first group (G1) includes the ideal, blurry, and defected plates. The second group (G2) contains double-row LPs, trees, similarity of LP and car body, and more than one LP or a car in a single image. The third group (G3) contains excessive light, rainy, shady, and low-contrast images. Table III shows the number of images per each classified group.

To show the increment percentage of the detection rate, Fig. 24 shows the detection rate of each group before and after applying (12). As shown from the results of this figure, the increment percentage of the detection rate after applying (12) has been raised from 86.3% to 91.65% for all test images. Fig. 25 shows the total number of success and failure detection for each group after applying (12). The detection rate for G2 and G3 is lower than G1 due to the complexity existing in the image. Table IV shows the percentage of success and failure for each group. Fig. 26 shows the average processing times for all test images. Most of the processed images take 47 ms. The computation time for each of the seven stages in the proposed method is listed in Table V. A lot of the time is consumed on the second stage, which is AT. The total time of processing one 352 $\times$ 288 image is 47.7 ms, which meets the real-time processing requirement. The stage that requires the longest time to compute is AT. The Combination of the ULEA and the VEDA still consumed less time than AT.

As summarized, 607 LPs from the 664 images are successfully detected. Then, the average accuracy of CLPD is 91.4%.

In addition to the successfully detected LPs shown in Fig. 23, a set of further samples that are taken randomly is shown in Fig. 27, and their located LPs are marked. It can be seen that the proposed method is able to detect LPs, although the LP is tilted. In addition, if there are many cars or LPs on a single image, the proposed CLPD method can detect them and extract the correct LPs, as shown in Fig. 27.

A comparison of the proposed CLPD method with several methods will be provided in Table VI. Additionally, in Table VII, the computation time of the proposed CLPD method is compared with the computation time of other methods.

As shown in Table VII, our proposed CLPD method can be used with real-time applications, and it has the lowest computation time compared with several methods.

3) CLPD Method Evaluation Using Sobel: In this case, the following steps have been used: ULEA→Sobel→HDD→CRE→PRS→PD. Table VIII summarizes the computation time and the detection rate of the proposed CLPD method.

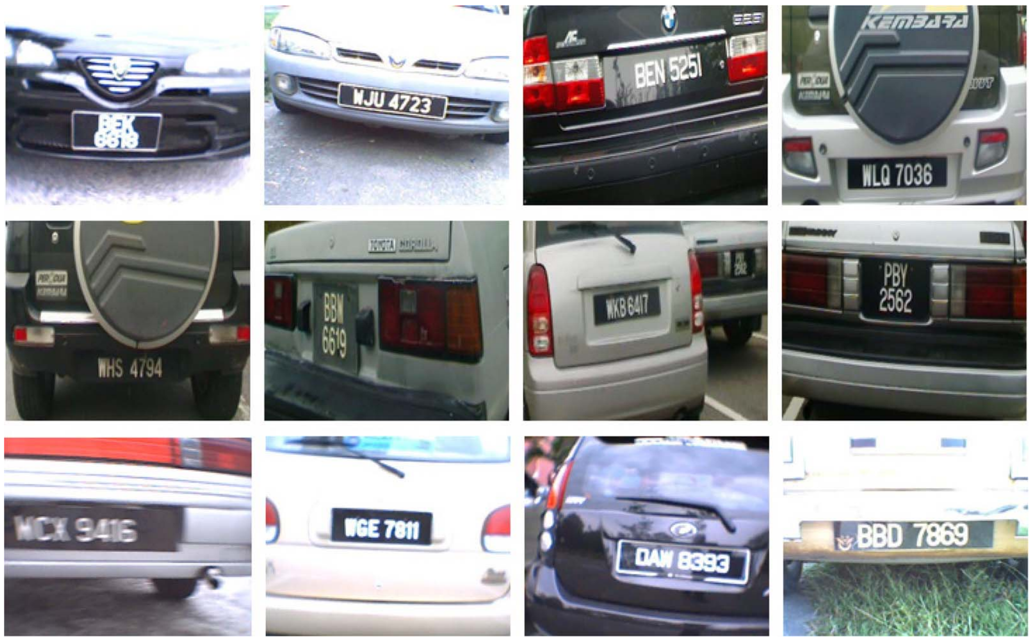

Fig. 23. Various captured LPs.

TABLE II
EXPERIMENTAL CONDITIONS

environment	outside
weather	rainy, sunny, shady days
capturing time	8am - 6pm
total number of plates	664
camera specs	30 fps
camera distance to LP	2 m - 4 m
camera placement	Pan ±20°, tilt 20°
image size	352×288
image background	complex; not fixed
license plates sizes	different
license plates designs	different; has multiple rows in a LP
colors of vehicles	various

TABLE III
EXPERIMENTAL CLASSIFICATION

No. group	#images
G1	176
G2	266
G3	222
Total	664

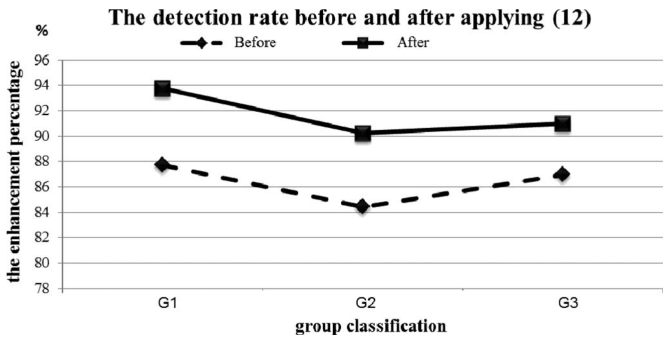

Fig. 24. Enhancement of the detection rate after using (12).

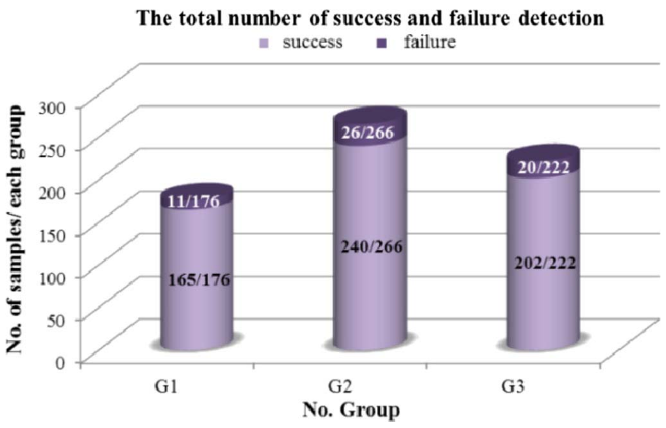

Fig. 25. Total number of success and failure detection.

TABLE IV
PERCENTAGE OF SUCCESS AND FAILURE DETECTION

Group	Detection rate
G1	93.75%
G2	90.23%
G3	90.99%
Averaged detection rate	**91.65%**

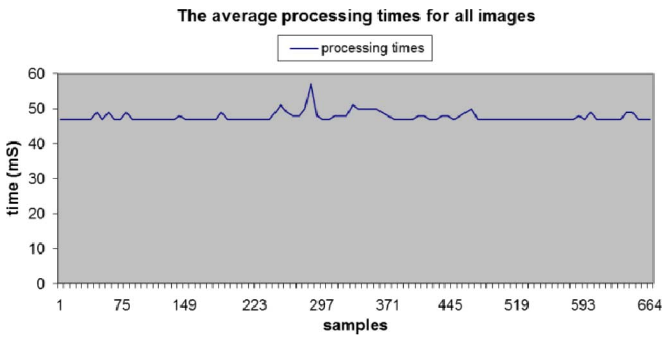

Fig. 26. Average processing times for all tested images.

TABLE V
COMPUTATION TIME OF EACH OF THE SEVEN STAGES IN THE CLPD METHOD (IN MILLISECONDS)

color → gray	AT	ULEA	VEDA	HDD	CRE	PRS and PD	Total time
1.1	15	5.2	7	7.9	3.3	8.2	47.7

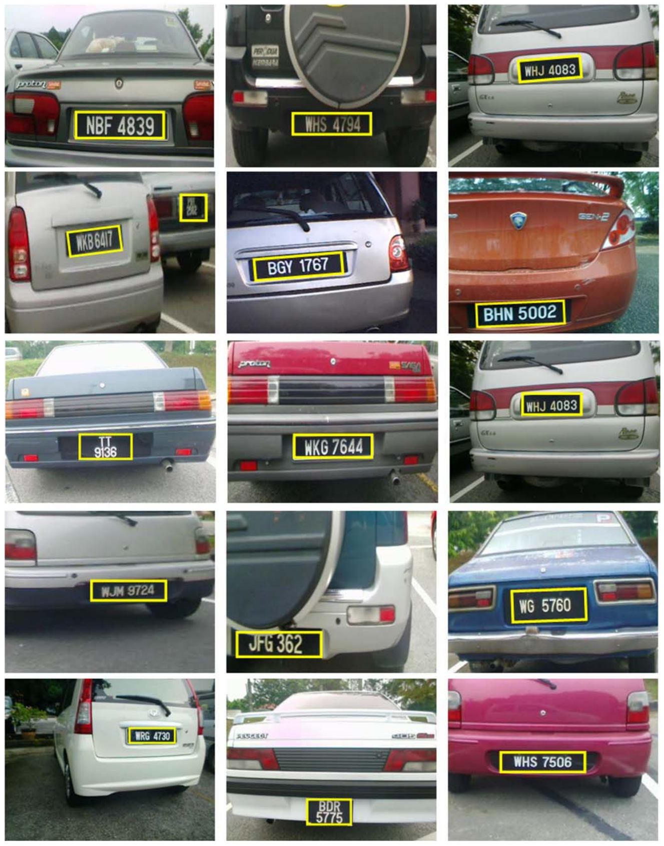

Fig. 27. Set of successfully detected LPs.

TABLE VI
COMPARISONS BETWEEN THE PROPOSED METHOD AND OTHER SEVERAL METHODS

[Reference]	Advantages	Shortcomings
[7]	• High efficiency; • Able to process rotated, low contrast LP	- Fixed sizes of LP are used.
[15]	• High efficiency; • Tolerance to rotation	- Complex background
[18]	• Tolerance to complex background, rotation, lighting, and low contrast	- High complexity
[20]	• Tolerance to lighting, low contrast, varied sizes of LP	- Complex background
[21]	• Good LP detection result; • Tolerance to lighting and low contrast	- Vulnerable to complex background and rotation
Proposed CLPD	• Tolerance to lighting, tilt, varied sizes and designs of LPs • Able to process complex background • Able to process low resolution image	- High complexity

TABLE VII
COMPUTATION TIME OF THE PROPOSED AND OTHER METHODS

[Reference]	Computation time (ms)	Detection rate (%)
[6]	1026	90
[18]	530	97.1
[20]	223	97.3
[21]	3000 (LPD and LPR)*	91.7
[25]	712	92.8
[42]	937	84%
Proposed CLPD	47.7	91.4%

TABLE VIII
COMPUTATION TIME AND DETECTION RATE OF THE CLPD METHOD USING VEDA AND SOBEL

Evaluation / Edge detector	No. correctly detected plates	Detection rate	Computation time (ms)
VEDA-based	607 / 664	91.4 %	47.7
Sobel-based	591 / 664	89 %	101.7

Based on these results, it is concluded that using the VEDA for detecting vertical edges could have enhanced the performance of the proposed CLPD in terms of the computation time and the detection rate.

VI. CONCLUSION

We have proposed a new and fast algorithm for vertical edge detection, in which its performance is faster than the performance of Sobel by five to nine times depending on image resolution. The VEDA contributes to make the whole proposed CLPD method faster. We have proposed a CLPD method in which data set was captured by using a web camera. We employed 664 images taken from various scenes and under different conditions. Only one LP is considered in each sample for the whole experiments. In the experiment, the rate of correctly detected LPs is 91.4%. In addition, the computation time of the CLPD method is 47.7 ms, which meets the real-time requirements. Finally, the VEDA-based and Sobel-based CLPD are compared, and the findings show that VEDA-based CLPD is better in terms of the computation time and the detection rate.

REFERENCES

[1] S. N. Huda, K. Marzuki, Y. Rubiyah, and O. Khairuddin, "Comparison of feature extractors in license plate recognition," in *Proc. 1st IEEE AMS*, Phuket, Thailand, 2007, pp. 502–506.
[2] S. Thanongsak and C. Kosin, "The recognition of car license plate for automatic parking system," in *Proc. 5th Int. Symp. Signal Process. Appl.*, Brisbane, QLD, Australia, 1999, pp. 455–457.
[3] H. Bai and C. Liu, "A hybrid license plate extraction method based on edge statistics and morphology," in *Proc. 17th Int. Conf. Pattern Recognit.*, Cambridge, U.K., 2004, pp. 831–834.
[4] M. Fukumi, Y. Takeuchi, H. Fukumoto, Y. Mitsura, and M. Khalid, "Neural network based threshold determination for Malaysia license plate character recognition," in *Proc. 9th Int. Conf. Mechatron. Technol.*, 2005, pp. 1–5.
[5] E. R. Lee, K. K. Pyeoung, and J. K. Hang, "Automatic recognition of a car license plate using color image processing," in *Proc. IEEE Int. Conf. Image Process.*, 1994, pp. 301–305.
[6] R. Parisi, E. D. Di Claudio, G. Lucarelli, and G. Orlandi, "Car plate recognition by neural networks and image processing," in *Proc. IEEE Int. Symp. Circuits Syst.*, 1998, pp. 195–198.
[7] T. Naito, T. Tsukada, K. Yamada, K. Kozuka, and S. Yamamoto, "Robust license-plate recognition method for passing vehicles under outside environment," *IEEE Trans. Veh. Technol.*, vol. 49, no. 6, pp. 2309–2319, Nov. 2000.
[8] H.-H. P. Wu, H.-H. Chen, R.-J. Wu, and D.-F. Shen, "License plate extraction in low resolution video," in *Proc. IEEE 18th Int. Conf. Pattern Recognit.*, Hong Kong, 2006, pp. 824–827.
[9] A. M. Al-Ghaili, S. Mashohor, A. Ismail, and A. R. Ramli, "A new vertical edge detection algorithm and its application," in *Proc. IEEE Int. Conf. Comput. Eng. Syst.*, Cairo, Egypt, 2008, pp. 204–209.
[10] J.-W. Hsieh, S.-H. Yu, and Y. S. Chen, "Morphology-based license plate detection from complex scenes," in *Proc. 16th Int. Conf. Pattern Recognit.*, Quebec City, QC, Canada, 2002, pp. 176–179.
[11] A. A. Lensky, J. Kang-Hyun, and V. V. Gubarev, "Vehicle license plate detection using local fractal dimension and morphological analysis," in *Proc. IEEE 1st Int. Forum Strategic Technol.*, Ulsan, Korea, 2006, pp. 47–50.

[12] F. Martin, O. Martin, M. García, and J. L. Alba, "New methods for automatic reading of VLP's (Vehicle License Plates)," in *Proc. IASTED Int. Conf. Signal Process. Pattern Recognit. Appl.*, Heraklion, Greece, 2002, pp. 126–131.

[13] S.-H. Le, Y.-S. Seok, and E.-J. Lee, "Multi-national integrated car-license plate recognition system using geometrical feature and hybrid pattern vector," in *Proc. Int. Tech. Conf. Circuits Syst. Comput. Commun.*, Phuket, Thailand, 2002, pp. 1256–1259.

[14] J. R. Parker and P. Federl, "An approach to license plate recognition," in *Proc. Visual Interface*, Kelowna, BC, Canada, 1997, pp. 178–182.

[15] M. Yu and Y. D. Kim, "An approach to Korean license plate recognition based on vertical edge matching," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, 2000, pp. 2975–2980.

[16] H. Zhang, W. Jia, X. He, and Q. Wu, "A fast algorithm for license plate detection in various conditions," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Taipei, Taiwan, 2006, pp. 2420–2425.

[17] D. Zheng, Y. Zhao, and J. Wang, "An efficient method of license plate location," *Pattern Recognit. Lett.*, vol. 26, no. 15, pp. 2431–2438, Nov. 2005.

[18] J.-M. Guo and Y.-F. Liu, "License plate localization and character segmentation with feedback self-learning and hybrid binarization techniques," *IEEE Trans. Veh. Technol.*, vol. 57, no. 3, pp. 1417–1424, May 2008.

[19] S. Kim, D. Kim, Y. Ryu, and G. Kim, "A robust license-plate extraction method under complex image conditions," in *Proc. 16th Int. Conf. Pattern Recognit.*, Quebec City, QC, Canada, 2002, pp. 216–219.

[20] Z.-X. Chen, Y.-L. Cheng, F.-L. Chang, and G.-Y. Wang, "Automatic license-plate location and recognition based on feature salience," *IEEE Trans. Veh. Technol.*, vol. 58, no. 7, pp. 3781–3785, Sep. 2009.

[21] H. Caner, H. S. Gecim, and A. Z. Alkar, "Efficient embedded neural-network-based license plate recognition system," *IEEE Trans. Veh. Technol.*, vol. 57, no. 5, pp. 2675–2683, Sep. 2008.

[22] S. Rovetta and R. Zunino, "License-plate localization by using vector quantization," in *Proc. Int. Conf. Acous., Speech, Signal Process.*, 1999, pp. 1113–1116.

[23] H. Bai, J. Zhu, and C. Liu, "A fast license plate extraction method on complex background," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, 2003, pp. 985–987.

[24] K. Debi, H.-U. Chae, and C. K.-H. Jo, "Parallelogram and histogram based vehicle license plate detection," in *Proc. IEEE Int. Conf. Smart Manuf. Appl.*, Gyeonggi-do, Korea, 2008, pp. 349–353.

[25] S. K. Kim, D. W. Kim, and H. J. Kim, "A recognition of vehicle license plate using a genetic algorithm based segmentation," in *Proc. Int. Conf. Image Process.*, Lausanne, Switzerland, 1996, pp. 661–664.

[26] B.-H. Ron and J. Erez, *A Real-Time Vehicle License Plate Recognition (LPR) System*, 2002. [Online]. Available: http://visl.technion.ac.il/projects/2003w24/

[27] O. Shahaf and J. Erez, *Real Time License Plate Recognition in a Video Movie*, 2002. [Online]. Available: http://visl.technion.ac.il/projects/2002w03/

[28] J. Matas and K. Zimmermann, "Unconstrained license plate and text localization and recognition," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, Vienna, Austria, 2005, pp. 572–577.

[29] S.-L. Chang, L.-S. Chen, Y.-C. Chung, and S.-W. Chen, "Automatic license plate recognition," *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 1, pp. 42–53, Mar. 2004.

[30] W. Jia, H. Zhang, and X. He, "Region-based license plate detection," *J. Netw. Comput. Appl.*, vol. 30, no. 4, pp. 1324–1333, Nov. 2007.

[31] R. C. Hardie and C. G. Boncelet, "Gradient-based edge detection using nonlinear edge enhancing prefilters," *IEEE Trans. Image Process.*, vol. 4, no. 11, pp. 1572–1577, Nov. 1995.

[32] S. Thanongsak and C. Kosin, "Extracting of car license plate using motor vehicle regulation and character pattern recognition," in *Proc. IEEE Asia-Pac. Conf. Circuit Syst.*, 1998, pp. 559–562.

[33] V. Abolghasemi and A. Ahmadyfard, "Improved image enhancement method for license plate detection," in *Proc. 15th Int. Conf. DSP*, 2007, pp. 435–438.

[34] L. S. Davis, "A survey of edge detection techniques," *J. Comput. Graph. Image Process.*, vol. 4, no. 3, pp. 248–270, Sep. 1975.

[35] J. K. Aggarwal, R. O. Duda, and A. Rosenfeld, *Computer Methods in Image Analysis*. New York: IEEE Press, 1977.

[36] S. Gendy, C. L. Smith, and S. Lachowicz, "Automatic car registration plate recognition using fast hough transform," in *Proc. IEEE 31st Annu. Int. Carnahan Conf. Security Technol.*, Canberra, ACT, Australia, 1997, pp. 209–218.

[37] D. Bradley and G. Roth, "Adaptive thresholding using the integral image," *J. Graph. Tools*, vol. 12, no. 2, pp. 13–21, Jun. 2007.

[38] F. Shafait, D. Keysers, and T. M. Breuel, "Efficient implementation of local adaptive thresholding techniques using integral images," in *Proc. Doc. Recognit. Retriev.*, 2007, pp. 681510-1–681510-6.

[39] P. D. Wellner, "Adaptive thresholding for the DigitalDesk," Rank Xerox Ltd., Birmingham, U.K., Tech. Rep. EPC-1993-110, 1993.

[40] F. Porikli and O. Tuzel, "Fast construction of covariance matrices for arbitrary size image windows," in *Proc. Int. Conf. Image Process.*, 2006, pp. 1581–1584.

[41] M. S. Nixon and A. S. Aguado, *Feature Extraction and Image Processing*, 1st ed. Woburn, MA: Reed, 2002, p. 106.

[42] C. C. Ming *et al.*, "ProjekCARPET, Car License Plate Extraction & Recognition Technology," presented at the National Innovation Summit, Kuala Lumpur, Malaysia, Apr. 29, 2004. [Online]. Available: http://www.projekcarpet.com.my/press77.html

Abbas M. Al-Ghaili received the B.Eng. degree (with first-class honors) in computer engineering from the University of Science and Technology, Sana'a, Yemen, in 2005 and the M.Sc. degree in computer systems engineering from the Universiti Putra Malaysia (UPM), Serdang, Malaysia, in 2009. He is currently working toward the Ph.D. degree in computer systems engineering with UPM.

His research interests include image processing, artificial intelligence, and software engineering.

Mr. Al-Ghaili is a member of the International Association of Computer Science and Information Technology and the Universal Association of Computer and Electronics Engineers.

Syamsiah Mashohor received the B.Eng. degree from the Department of Computer and Communication Systems Engineering, Universiti Putra Malaysia (UPM), Serdang, Malaysia, in 2002 and the Ph.D. degree from the University of Edinburgh, Edinburgh, U.K., in 2006.

Since 2002, she has been with the Department of Computer and Communication Systems Engineering, UPM, where she is currently a Senior Lecturer. She has taught a range of computing topics and is particularly interested in teaching programming and image processing. Her research interests include artificial intelligence, particularly genetic algorithms, and image processing.

Abdul Rahman Ramli received the B.S. degree in electronics from the National University of Malaysia, Bangi, Malaysia; the Master's degree in information technology systems from the University of Strathclyde, Glasgow, U.K.; and the Ph.D. degree from the University of Bradford, Bradford, U.K.

From August 1996 to July 1998, he was a Department Head with the Department of Computer and Communication Systems Engineering, Universiti Putri Malaysia, Serdang, Malaysia. His research interests include image processing and electronic imaging, multimedia systems engineering, Internet computing, smart card applications, embedded systems, computer remote monitoring systems, and intelligence systems.

Alyani Ismail received the B.Eng. (Hons) degree in electronics and information engineering from the University of Huddersfield, Huddersfield, U.K., and the M.Sc. degree in communication, computer, and human-centered systems and the Ph.D. degree in microwave devices from the University of Birmingham, Birmingham, U.K.

From 2010 to February 2012, she was a Department Head with the Department of Computer and Communication Systems Engineering, Universiti Putra Malaysia (UPM), Serdang, Malaysia. She is currently an Associate Professor with the Department of Computer and Communication Systems Engineering, UPM. Her research interests include microwave devices, micro/nanodevices, and detection systems.