Zalili Binti Musa; Junzo Watada

A grid-computing based multi-camera tracking system for vehicle plate recognition

Kybernetika, Vol. 42 (2006), No. 4, 495--514

Persistent URL: `http://dml.cz/dmlcz/135731`

KYBERNETIKA — VOLUME *42* (2006), NUMBER 4, PAGES 495 – 514

A GRID–COMPUTING BASED MULTI–CAMERA TRACKING SYSTEM FOR VEHICLE PLATE RECOGNITION

Zalili Binti Musa and Junzo Watada

There are several ways that can be implemented in a vehicle tracking system such as recognizing a vehicle color, a shape or a vehicle plate itself. In this paper, we will concentrate ourselves on recognizing a vehicle on a highway through vehicle plate recognition. Generally, recognizing a vehicle plate for a toll-gate system or parking system is easier than recognizing a car plate for the highway system. There are many cameras installed on the highway to capture images and every camera has different angles of images. As a result, the images are captured under varied imaging conditions and not focusing on the vehicle itself. Therefore, we need a system that is able to recognize the object first. However, such a system consumes a large amount of time to complete the whole process. To overcome this drawback, we installed this process with grid computing as a solution. At the end of this paper, we will discuss our obtained result from an experiment.

Keywords: vehicle plate recognition, grid computing, recognition system, tracking system

AMS Subject Classification: 68G35, 68U10

1. INTRODUCTION

Today it is important to recognize car behaviors, human behaviors and products, etc. through many cameras in security systems, inventory systems, production systems and so on. In the present state of arts, many cameras are employed to review, watch and recognize irregular or unsafe behaviors. For example, it is not easy to track a person's or a car's behavior over many cameras, each of which consists a limited partial space in a vast universe. Therefore, it is required to automatically track a human or vehicle behavior over a multi-camera view.

In this paper, we will discuss a multi-camera tracking system, where we place a stress on a highway through vehicle plate recognition. Generally, recognizing a vehicle plate for a toll-gate system or parking system is easier than recognizing a car plate for a highway system. There are many cameras installed on the highway to capture images and every camera has different angles of images. As a result, the images are captured under varied imaging conditions and not focusing on the vehicle itself. Therefore, we need a system that is able to recognize the object first. However, such a system consumes a large amount of time to complete the whole process. To overcome this drawback, we installed this process with grid computing as a solution. And a brief future view of a multi-camera tracking system is given.

In the reminder of this paper, we will explain our motivation briefly in Section 2 and literature review in Section 3. Then, in Section 4, we describe the architecture the system included formula and algorithm that we used in our method. Finally, we conclude with a discussion and opportunities for future work in Section 5 and Section 6.

2. MOTIVATION

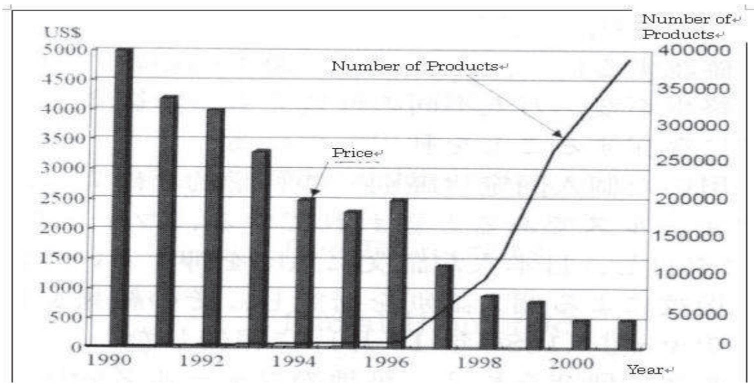

Fig. 1. Price and Product Amount of Biometric Apparatus.

A multi-camera tracking system has a wide possibility in a today's society. This system is dealt in biometrics [28] and situation/context awareness technologies [27]. Today biometric technology is widely used in a real world. The production of biometric apparatuses increases exponentially from 1996 as shown in Figure 1. (This figure is cited from [28]). Contrarily, the price of each apparatus decreases linearly each year. According to the increase of its use, the price in 2000 is one tenth less than one in 1990. This means rapidly such technologies invaded in the society.

But such researches have very partly and individually been pursued. In order to detect an unsafe situation, it is required to recognize a natural situation, environment or human and vehicle behaviors in a whole time-length and in a spatial universe. One method is to provide a tracking system employing views taken by cameras scattered and placed in various positions or from various angles in buildings, campus, airports, stations, streets, highways, or towns for tracing environments, vehicles and humans.

However, the creation, processing and management of these data types require an enormous computational effort, often too high for a single processor architecture. Therefore, we need a parallel and distributed computing in oder to speed up the process of image and video application as a solution.

3. RELATED WORKS

The objective of this section is to provide the present state of arts and discuss a future trend to researches on a multi-camera tracking system. In the literature, three main kinds of multi-camera tracking system are outlined. The first rely on multi-camera tracking system, secondly rely on localization of vehicle on vehicle tracking system on the road and the thirdly rely on plate location detection.

3.1. Multi-camera tracking system

Andrew D. Bagdanov et al. employ coordinated camera networks to acquire highly resolution images of many moving targets [3]. Shingo Uchihashi discusses camera control interfaces in order to capture meetings [30]. Jun Wang et al. discuss experimental sampling method for video surveillance [31]. Due to the decreasing costs and increasing miniaturization of video cameras, the use of digital video based surveillance as a tool for real-time monitoring is rapidly increasing. Zohar Naor develops a wireless dynamic tracking method [23]. H. G. Adshead discusses some known automatic tracking techniques [1]. Andreas Krause et al. discuss data association and intensity tracking of multiple topics over time based on factorial hidden Markov models [14].

Regarding human behavior tracking, Daniel Gatica-Perez et al. deals with tracking a meeting conversation [11]. Andrea Prati et al. deal with enhancing video surveillance systems. Multi-modal sensor integration can be a successful strategy [25]. In this work, a computer vision system able to detect and track people from multiple cameras is integrated with a wireless sensor network mounting PIR (Passive InfraRed) sensors. Yuan-Fang Wang et al. propose a system based on a distributed, multi-camera video analysis for airport security surveillance [31]. It uses biometric signatures, which are called soft biometry including a person's height, built, skin tone, color of shirts and trousers, motion pattern, trajectory history, etc., to ID and track errant passengers and suspicious events without having to shut down a whole terminal building and cancel multiple flights. Douglas Fidaleo et al. deal with analysis of facial gestures through face recognition [9]. They employed facial gestures to understand the meaning of a face. Tarak Gandhi et al. analyze three dimensional calibration using multi-cameras [10]. Joel Carrnza et al. provide a model of 3-D space dynamic real-world scene [6]. They exploit robust motion estimation of a human body and its motion. Xiaocong Fan et al. provide reasoning method for tracking a team [8] by means of agent in a dynamic real time domains [8].

3.2. Localization of vehicle

Nowadays, a tracking system is an important research in a security system, manufacturing, medical and automatic traffic monitoring. An automatic traffic monitoring system plays a central role in intelligent transportation systems (ITS). Various sensors and magnetic detection loop have been applied but the installation and maintenance of loop detectors caused inconvenience and excessive cost. Therefore, image processing and computer vision techniques have been studied to provide a more powerful solution [18].

As we know, one of the most important tasks in a vehicle tracking system is a localization of preceding a vehicle [15, 18]. In late 1980s, many tracking algorithms are focused on tracking with a fixed camera position and late of 1990s, there are approaches integrating wide-area scenes with target tracking with mobile camera [29]. These, required more than one camera and more computing resource to execute the system. According to Sholin Kyo et al. used an edge-based method to search the location of a vehicle and for a real-time implementation, this system is executed on IMAP-Vision, a PCI bus image processing board is used together with a normal PC [15].

Meanwhile, Ching-Po et al. used a counter growing method and a real-time image processing card, Flex10K200s from Altera to solve the initialization problem of a multi-lane artery image tracking. In there, active counters are adopted to represent the images of vehicles and a dynamic model is proposed to predict vehicle position in the image sequence for a vehicle tracking system [18].

Based on a vehicle queue detection system, Michele Zanin et al. used an edges detection method to detect the location of a vehicle. In there, this system is executed on Pentium III 700 MHz PC with 256MB Ram. Its acquisition device is a Hauppauge WinTv PCI card and operation system is Linux [20].

Generally, many of researchers have the own methods to solve the problem of localization of a vehicles, for examples they used a neural network [29], a three-dimensional(3D) model-based method [13], a region-based method [12], a feature-based method [5] and a transformation based method [21]. But their papers did not discuss about computing resources to execute their system.

3.3. Vehicles plate location detection

As we mentioned above, in this paper we concentrated on recognizing a vehicle on a highway through vehicle plate recognition. Basically, plate location detection is the most important part in an automated recognition system of a car plate. Failure to detect the location of the plate will eventually fail the entire system [2, 7, 16, 19].

In 90's, a statistical analysis method is a common one used for the detection of a car plate location. Many of researchers agreed that a statistical analysis method provided a better solution. For example, Sorin Draghici used a histogram technique to detect the plate location [7]. Meanwhile, M. A. Alias used variation techniques for car plate location detection [2]. Both Sang et al. and Lim et al., used Hough Transform to detect a plate location [17, 26], Barroso et al. using statistic analysis to capture feature of a car plate location [4].

In the late 90's transformation technique has been used. According to Fernando Martin and David Borges [19], they employed morphological transformation as a method to perform plate location. Meanwhile in 2002, Byongmo Lee and Euiyoung Cha applied the Fast and Robust techniques to detection of car plate using HSV and Weighted morphology [16].

In this paper, we are focusing on recognizing a vehicle on a highway through vehicle plate recognition. The new algorithm for localization of vehicle and car plate location detection will be discussed further in this paper.

4. SYSTEM ARCHITECTURE

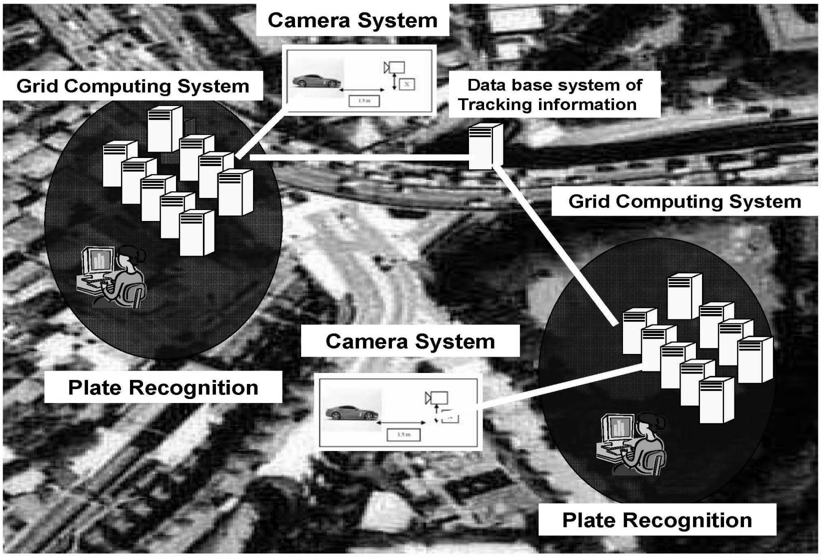

Fig. 2. Multi-camera tracking system based on grid computers.

In this system, we construct a multi-camera tracking system that consists of many cameras, a grid computer system, a database and a network. Many cameras build a vast view universe consisting of some limited partial spaces. The grid computer system provides with a car plate recognition system. The database provides a linked information obtained through many cameras and supports the car plate recognition system. The network connects all grid computer subsystems with the database system [22, 32] as shown in Figure 2.

In order to realize the scenarios above, a framework is needed to ensure its quick processing for localization of a vehicle and detecting a plate location shown in Figure 4. Generally, this method consists of four main module: 1) image acquisition, 2) pre-processing 3) vehicle location detection and 4) vehicle plate location detection.

The objective of the image acquisition module is to capture an image from a video digital camera and send it to pre-processing module for the next process. In the pre-processing module, the RGB images are converted to gray level and at the same time, the process of removing noise and unknown object is executed. These two process are very important to ensure the process of vehicle location detection can execute smoothly. After we obtained the location of a vehicle, the process of plate location detection is executed. In oder to distribute the process into several processors, we used AD-POWERs as a grid application from Dai Nippon Printing Co. Ltd. (DNP) as a tool.

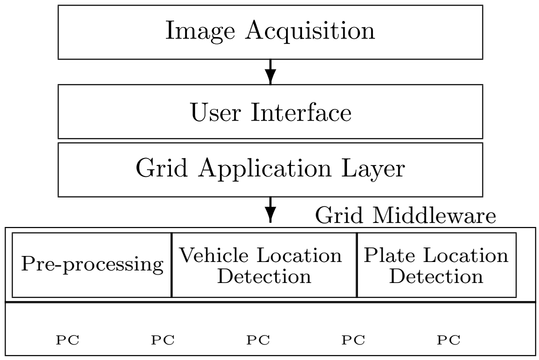

Fig. 3. Multi-camera tracking system based on grid computers.

4.1. Image acquisition

Fig. 4. An example of data image.

In this research, images of car plates are obtained through a Cannon digital camera with 5.0 Mega pixels with zooming capability. The cameras were positioned on three different areas at Fukuoka Kitakyushu Urban Expressway. For each area, about 30 vehicle images have been captured and the size of each image is 640×480 pixels in JPEG format. Figure 4 shows an example of an image.

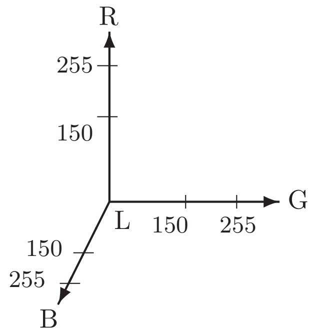

Fig. 5. Relationship between RGB and brightness.

4.2. Pre-processing

The pre-processing module consists of two processes which provide threshold and filtering. The purpose of threshold process in this stage is to simplify the positioning of a vehicle based on RGB colors and brightness. Meanwhile, the purpose of filtering process is to remove unknown objects and also to convert the RGB images into a binary image.

As we know highways, they have more than two lanes where vehicles flow. Therefore, it is important for the system to recognize the location of a vehicle first.

In this research, image color has been used. Based on our studies we identified that the color of the road and vehicle's shadows are important features in the threshold process. Therein, we find that the major color of the road is blue and also that the vehicle's shadows have a certain value of the brightness. Assuming $M \times N$ size image I, therefore denoting an image in red, green or blue is respectively as:

$$I(R, G, B)$$

where

$$(0 \leq R(x,y), G(x,y), B(x,y) \leq 255).$$

Based on the above equation, x and y denote spatial coordinates and the value of $I(N, M)$ at any point, where x is a row number and y is a column number. So (x, y) is proportional to the brightness of the image at that point. Figure 5 shows the relationship between RGB colors and brightness.

In this process the value of a gray scale is denoted by δ. As we mentioned above the process to convert the RGB images into a gray level depends on numbers of blue color, that denote $B_{(c)}$ where the value of a blue color is greater than the value of a red or a green color on a RGB image. There is the calculation of $B_{(c)}$ and δ, for each pixel is expressed as:

$$B_{(c)} = \begin{cases} B_{(c)} + 1 & : ((B(x,y) > R(x,y)) \ \& \ (B(x,y) > G(x,y))) \\ B_{(c)} & : \text{ otherwise} \end{cases} \tag{1}$$

$$\delta = 255 - (R(x,y) + G(x,y) + B(x,y)). \tag{2}$$

Based on Eq. (1) and Eq. (2) formula to convert the RGB into gray level image $g(x,y)$, it is defined as follows:

$$g(x,y) = \begin{cases} 0: & B_{(c)} \leq 20 \\ \delta: & B_{(c)} > 20. \end{cases} \tag{3}$$

Assuming $M \times N$ size image, therefore, the threshold process is summarized as in the following algorithm:

i. Divide the image into small windows
ii. For each pixel on smaller windows
Get the $B_{(c)}$ on the images as Eq.(1)
Get the value of a gray level,δ as Eq.(2)
iii. If ($B_{(c)} > 20$)
iv. $g(x,y) = \delta$
v. Else $g(x,y) = 0$
vi. Repeat steps i until v for whole images.

According to algorithm above, Figure 6 shows an image after the threshold process.

Fig. 6. An image after the threshold process.

4.2.1. Filtering

As we mentioned above, the second process is to remove noise and also to convert the gray image into a binary image. After the threshold process, we observed the images that still have noises and unknown objects. For example, landscape objects and vehicles these are not so clear. So that we employed the formula as shown in Eq. (4) for converting the gray image into a binary image.

$$\lambda = 0.75\ \omega \tag{4}$$

where λ is a threshold value and ω is the maximum value of the image. Based on Eq.(4) formula to convert the gray image to binary image $d(x,y)$, it is defined as follows:

$$d(x,y) = \begin{cases} 1: & g(x,y) > \lambda \\ 0: & \text{otherwise.} \end{cases} \tag{5}$$

Therefore, the algorithm to convert the gray level into a binary image is written as follows:

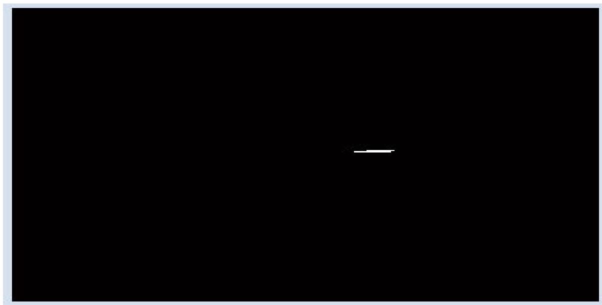

Fig. 7. An image after the filtering process.

i. Get the maximum value of overall images, ω
ii. Get the threshold value of images, λ as Eq.(4)
iii. Loop $x = 1$ to M
iv.   Loop $y = 1$ to N
v.   If $g(x,y) > \lambda$
vi. $d(x,y) = 1$
vii.   otherwise, 0.

Figure 7 shows an image after the filtering process.

4.3. Vehicle location detection

After performing the threshold and filtering, the detection process of a vehicle location can start. In this stage the features of the images are more prominent to our naked eyes and the algorithm will be applied here. The algorithm of the vehicle location detection is shown as follows:

i. Loop $x = 1$ to M
ii.   Loop $y = 1$ to N
iii.   Read pixel of image, $d(x,y)$
iv.   If $d(x,y) = 1$
v.   Get the length of pixel value 1, ℓ
vi. $\alpha = x - ((y - \ell)/2)$
vii. Get the location of vehicle images on gray scale, $g(x:\alpha, y:\ell)$
viii. Repeat step i until vii for whole images.

Figure 8 shows the image of a vehicle location.

Fig. 8. An image of a vehicle location.

4.4. Vehicle plate location detection

Basically, the vehicle plate location detection module consists of two major processes which are 1) analysis of a pattern based on a horizontal line and 2) analysis of a pattern based on a vertical line. Both processes consist of three different subprocesses which are clustering, filtering and location detection.

4.4.1. An analyzed pattern based on a horizontal line

In this process, there are three subprocesses in analyzing a pattern based on a horizontal line, which are clustering, filtering and location detection. The purpose of clustering process is to cluster spectrum data based on a histogram plotted. Meanwhile, filtering process is to simplify the pattern. Therefore, it is easy to detect the location of a vehicle plate.

- **Clustering**

In this process Fast Fourier Transform (FFT) based on Successive Doubling [24] is applied. The formula for FFT is written as:

$$F_{\text{even}}(u) = \sum_{x=0}^{M-1} f(2x) W_M^{ux} \tag{6}$$

$$F_{\text{odd}}(v) = \sum_{x=0}^{M-1} f(2x+1) W_M^{vx} \tag{7}$$

where $u = 0, 1, 2, \ldots, M-1$ and $v = 0, 1, 2, \ldots, M-1$. Meanwhile, a formula to cluster based on a histogram shows as below:

$$S(i) = \begin{cases} 0 : & (\gamma < \epsilon) \ \ \& \ \ (\gamma < \rho) \\ S(i) : & Otherwise \end{cases} \tag{8}$$

where γ is a current value, ϵ is a previous value and ρ is a next value of spectrum images, represented by $S(i)$. Meanwhile, i is the row number of a spectrum image.

6.5850	6.5880	6.6090
-0.7892 - 1.4467i	-0.7902 - 1.4382i	-0.7804 - 1.4294i
1.2393 - 1.3907i	1.2289 - 1.4096i	1.2214 - 1.4323i
-0.3784 - 0.5120i	-0.3954 - 0.4957i	-0.4138 - 0.4618i
-0.1780 + 0.0499i	-0.1555 + 0.0500i	-0.1153 + 0.0616i

Fig. 9. Spectrum values.

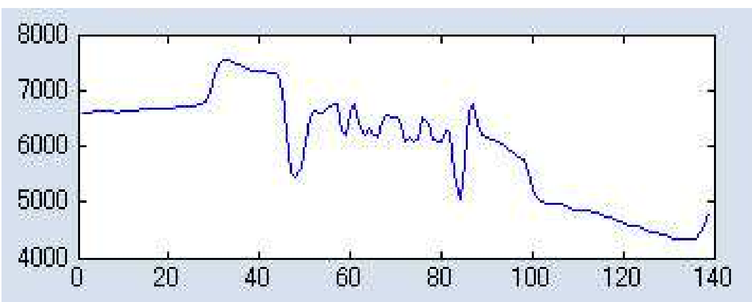

Fig. 10. Spectrum data $\sum_{x=0}^{M-1}$ plotted.

Based on the equations above, the algorithm of the clustering process is shown as follows:

i. Read the pixels of an image
ii. Transform them by FFT based on Eq.(6) and Eq.(7).
From these equations a number of spectrums will exit. Figure 9 shows an example of spectrum.
iii. Read the first line of a spectrum.
According to this value, Figure 10 shows a graph that spectrum data are plotted.
iv. Loop $i = 1$ to M
v. If $i = 2$
vi. Read γ, ϵ and ρ.
vii. If $(\gamma < \epsilon)$ and $(\gamma < \rho)$ then
viii. $S(i) = 0$
ix. Otherwise, keep the existing value of $S(i)$
x. Repeat steps i until ix for whole spectrum values

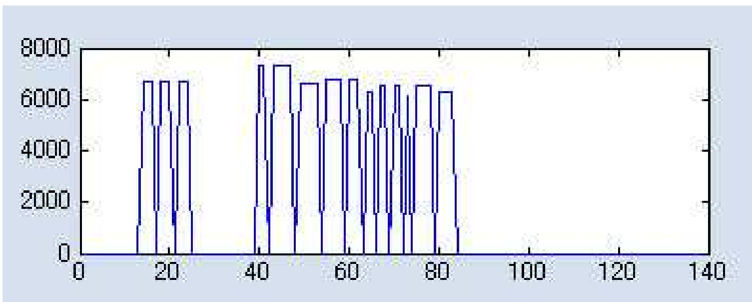

Fig. 11. Result of histogram after filtering process.

- **Filtering**

As we mentioned before, the purpose of filtering is to simplify a pattern. Therefore, the algorithm of the filtering process is denoted as follows:

i. count $= 0$
ii. loop $i = 1$ to M
iii. if $S(i) > 0$
iv. count $=$ count $+ 1$
v. else if (count > 8)
vi. $t = i -$ count
vii. convert $S(i)$ to $S(t) = 0$
viii. Repeat steps i to vii for whole spectrum values

Based on the algorithm above, Figure 11 shows a result of the histogram after filtering process.

- **Location Detection**

In this stage the pattern of the image enables us to recognize the location of a vehicle plate based on a horizontal line. The algorithm of the location detection is shown as follows:

```
i.     count = 0, max = 0
ii.    loop i = 1 to M
iii.       if S(i) > 0
iv.            count = count + 1
v.         else if (max < count)
vi.            max = count
vii.           count = 0
viii.  end loop
ix.    loop i = 1 to M
x.         if S(i) > 0
xi.            count = count + 1
xii.       else if(max = count)
xiii.          X = i and Υ = i − count
xiv.           Get the location of a vehicle plate
               on RGB images, I(X : Υ, 1 : N)
xv.    end loop
```

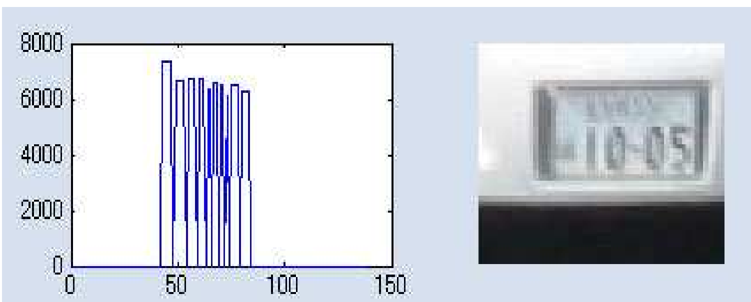

Fig. 12. Result of the location detection of a plate number based on an analyzed pattern on a horizontal line.

Figure 12 shows the result of the location detection process.

4.4.2. An analyzed pattern based on a vertical line

There are three colors in Japan car plates that is yellow, white and dark green. In our studies we identified that the green color of pixels is dominant to yellow, white and dark green. Therefore, we used only green color value to recognize the pattern of a location vehicle plate.

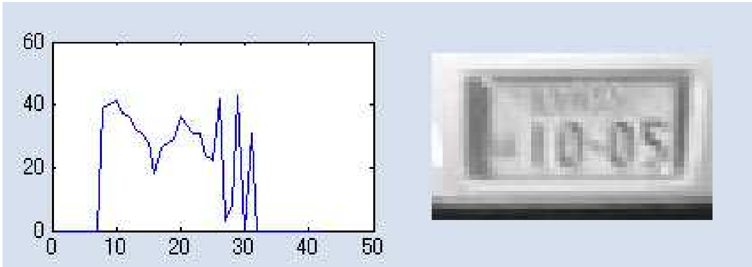

Fig. 13. Result of location detection of a plate number based on an analyzed pattern on a vertical line.

As we mentioned above, in analyzing a pattern based on a vertical line consisted of three processes: 1) clustering 2) filtering and 3) location detection. However we employed the same process as in analyzing a pattern based on a horizontal line for filtering and location detection. But for clustering process we applied a different technique from analyzing a pattern based on a horizontal line. In this situation, we used binary data that we converted from the value of a green color. The equation for converting the value of a green color to binary data is:

$$d(x,y) = \begin{cases} 0: & (200 \leq G(x,y) \leq 100) \\ 1: & (200 > G(x,y) > 100). \end{cases} \tag{9}$$

After converting the image to binary data through a threshold, the process of calculating the sum of value 1 in every row is executed. Then we used the same process to cluster the data based on histogram plots as the algorithm in Section 3.4.1. The result of this process is shown in Figure 13.

5. RESULT AND DISCUSSION

The proposed method and algorithm were developed, tested and applied. In order to evaluate the proposed method, 30 images of car plates have been tested under different conditions, which are:

- Vehicle images with different angles and sizes of a car plate.
- Vehicle images with another character and stickers near the plate.
- Images vehicles under a normal condition.

A prototype was developed using the proposed method and algorithm. The 30 images of car plates are tested. Based on the test, 96.3 % vehicles were detected correctly. Table 1 shows the result from the testing phases using 30 different images.

Table 2 shows the performance of a grid computing system employed, where the master and slave computers are a Dell compatible Pentium D 2.80GHz. As we

Table 1. Summary of testing.

No	Testing Data	Value	(%)
1.	Successful Plate Detection	30 / 30	100.0
2.	Failed Plate Detection	0 / 30	0.0
3.	Correct Location of Plate	29 / 30	96.3
4.	Incorrect Location of Plate	1 / 30	3.7

Table 2. Performance of grid computing system to process the whole images.

Process	Time Taken (second)
Pre-processing	120
Vehicle location detection	12
Plate location detection	6
Idle process	19

mentioned above, the highway has 2 lanes in one direction and considered the car speed is between 80 to 100 km/h. In this case, the grid computer system is employed where 5 slaved computers have been applied. Basically, one computer can pursue 88.5 % of one stand-alone. Table 2 and Table 3 shows the result of a speed test with 30 different vehicles.

In this case, the camera is taken 30 frames/second and for one image we take 23 second approximately to complete the whole process using this method. Regarding on Table 2 and Table 3 we found that one camera taken 4500 frames to record 30 different vehicles on the road. But we skip 145 frames (equivalent to 4.8 second) because these frames have almost the same vehicle view. Therefore, using a single processor we need 829.2 seconds to process the 30 images with different vehicles. Meanwhile, using grid computing with 5 slave and 2 backup computers the result shows that only 157 seconds is taken to complete the whole process.

Regarding the testing phase, a number of constraints were faced during the development of the Car Plate Detection System. The main constraint was the difficulty in detecting the location of a vehicle. This is because the vehicle streaks past very

Table 3. Comparison of processing time.

Speed km/h	80 – 100
30 cars passing a position per second	150.0
Processing time of one computer (s)	829.2
Processing time of 5 slave computer (s)	157.0

Table 4. Comparison of performance (1).

Method	Frame/second	Vehicle speed(km/h)	Vehicles localization
Edge based	15	none	96 %
active contour model	none	43.9 – 55.6	95.3 %
Vehicle shadows based	30	80 – 100	100 %

Table 5. Comparison of performance (2).

Method	Parking System	Vehicles traveling
Histogram	99 %	none
Coefficient of variation	86 %	none
Morphological transformation	85 %	none
Shape Based	100 %	99 % , 88.5 %
Fast Fourier Transform (proposed method)	none	100 %

fast. Therefore, the image is not so clear during the image acquisition process. The poor images also effected the results during the detection process of a plate number.

Regarding on Table 5., the comparison of several methods is discussed. The comparison result was tested under a normal weather condition. The result shows that our method can recognize the vehicle location 100 % compared with other methods. Where the result shows edge detection method and active contour model can detect vehicle location 96 % [15] and 95.3 % [18] respectively.

Table 5 shows the result of comparison of plate location detection with different methods. Based on histogram method, this method is able to recognize about 99 % of the location of registered plate for parking lot system [7]. However in this method, the researcher had to fix their car image position to the system requirement. Approximately, about 86 % was successful to detect the location of a car plate using coefficient of variation method [2]. However, this method has been performed at parking system and the position to image a car has been fixed as well. Meanwhile, based on morphological transformation in a plate location detection, the result of this method shows that about 85 % of the registered plates for a parking lot system has been recognized [19].

Regarding on shape based method, three image databases have been created to evaluate the proposed methods which are: Database 1, of images taken at the entrance of a car park; Database 2, of images of vehicles traveling on a road under a normal weather condition; and Database 3, of images of vehicles traveling on a road under a bad weather condition. The result shows, the system can detect a number plate location 100 % , 99 % and 88.5 % for Database 1, Database 2 and Database 3, respectively [4]. But their paper did not open the numbers of images they tested and also did not stated the condition of captured images.

Table 6. Comparison of performance (3).

Processor	Image attribute	Processing time
Pentium III	80 × 120 gray levels	8.4 ms/frame
IMAP-VISION	256 × 240 gray levels	65.0 ms
Grid Computing	640 × 480 RGB color	157 s

Based on our test, the result shows 100 % vehicles plate were detected correctly and approximately, about 96.7 % successful to detect the location of a vehicle plate. This result shows that one vehicle plate failed to locate on the correct location. This is because the vehicle streaks past very fast. Therefore, the image is not so clear during the image acquisition process. The poor images also effected the result during the detection process of a plate number.

Based on Table 6, the result shows processing time using different methods applied to different processors. According to Michele Zanin et al. [20], they have used shadow of a vehicle image for localization with size 80 × 120 gray levels. This work has tested using Pentium III 700 MHz with 256 MB RAM and the operation system is Linux. The result shows that the average computing time per frame is about 8.4 ms approximately. Meanwhile, based on Sholin Kyo et al. [15], they have used edge based method using a gray level image sized 256 × 240 to recognize the vehicle location. Their work has tested using an IMAP-VISION image processing board and C language for a linear processor array system. Where twenty potential vehicles have been checked. The result shows that the process has taken 65.0 ms to complete the whole process.

In our case, we used shadows based method using RGB image with 640 × 480 size image. We have tested our method using MatLab and C languages as operation system and grid computing with 5 slave and 2 backup computer. The result shows that only 157 seconds is taken to complete the whole process for thirty potential vehicles have been checked.

6. FUTURE VIEW

We presented the multi-camera tracking system for vehicles in a road. Such tracking systems are connected in a network and whole the obtained information is analyzed and reasoned into understanding situations. That is, people and cars in a street are tracked through video-cameras. In the future the system is able to recognize people and cars in a street. But such a human behavior does not have any plate number. Persons should be recognized in terms of their features such as a color, height, cloth patterns, cloth colors, and behavior patterns. Such information is accumulated in a database. The person is tracked beyond a single camera and captured in a universal dimensional space. As well, such information is linked with vehicle tracking systems for surveillance and security of a city [32].

7. CONCLUDING REMARKS

As shown in the first section, the Multi-camera Tracking System is constructed based on many cameras on a road such as highways. The obtained images are analyzed in high speed using a grid computing system. So one image of a camera view is dispatched to one of slave grid computers and analyzed. The obtained number of each car is transmitted to the data base system with a location and time. All numbers detected at each camera are linked in the database system. Then all cars are tracked on a highway. This system can be applicable to as a security system or as an automated toll system as shown in Figure 2.

As we presented the result in the previous section, we believed that our proposed method is capable to detect the location of a car plate as long as the car follows Japan's standard car plate specification. In the real environment of a vehicle tracking system on a highway the time of processing is a crucial issue. So that we suggest implementing this system in a grid computing environment in order to speed up the process.

The method based on vehicles shadow and Fast Fourier Transforms to detect a plate location of a car is proposed successfully. In testing the method we obtain the sufficiently good result.

ACKNOWLEDGEMENT

The first author would like to thank to Kolej Universiti Kejuruteraan dan Teknologi Malaysia (KUKTEM) and Kementerian Pengajian Tinggi Malaysia for a study leave.

(Received December 30, 2005)

REFERENCES

[1] H. G. Adshead: Optimising automatic tracking of multilayer boards. ACM SIGDA Newsletter *5* (1975), 3, 14–28.
[2] M. A. B. Alias: Pengesanan Kedudukan Nombor Plat Kereta Menggunakan Pendekatan Pekali Variasi. Bachelor Thesis, Universiti Teknologi Malaysia 1999.
[3] A. D. Bagdanov, A. del Bimbo, and F. Pernici: Explore multi-resolution views with PTZ and coordinated camera networks: Acquisition of high-resolution images through on-line saccade sequence planning. In: Proc. Third ACM Internat. Workshop on Video Surveillance & Sensor Networks VSSN'05. ACM Press, 2005, pp. 121–130.
[4] J. A. Barroso, A. Rafael, E. L. Dagless, and J. Bulas-Cruz: Number plate reading using computer vision. In: Proc. Internat. Symposium on Industrial Electronics (ISIE-97), Guimaraes 1997, pp. 761–766.
[5] D. Beymer, P. McLauchlan, B. Coihan, and J. Malik: A real-time computer vision system for measuring traffic parameters. In: Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition 1997, pp. 495–501.
[6] J. Carranza, Ch. Theobalt, M. A. Magnor, and H.-P. Seidel: Free-viewpoint video of human actors. ACM Trans. Graphics *22* (2003), 3, 569–577.
[7] S. Draghici: A neural network based artificial vision system for licence plate recognition. Internat. J. Neural Systems *8* (1997), 1, 113–126.
[8] X. Fan, D. Xu, J. Hou, and G. Zheng: Reasoning about team tracking. ACM SIGSOFT Software Engineering Notes *23* (1998), 3, 79–82.

[9] D. Fidaleo and M. Trivedi: Manifold analysis of facial gestures for face recognition. In: Proc. 2003 ACM SIGMM Workshop on Biometrics Methods and Applications, ACM Press 2003, pp. 65–69.

[10] T. Gandhi and M. M. Trivedi: Calibration of a reconfigurable array of omnidirectional cameras using a moving person. In: Proc. ACM 2nd International Workshop on Video Surveillance & Sensor Networks, ACM Press 2004, pp. 12–19.

[11] D. Gatica-Perez, G. Lathoud, J.-M. Odobez, and I. McCowan: Recognizing communication patterns: Multimodal multispeaker probabilistic tracking in meetings. In: Proc. 7th International Conference on Multimodal Interfaces ICMI'05, ACM Press 2005, pp. 183–190.

[12] M. Kilger: A shadow handler in a video-based realtime traffic monitoring system. In: Proc. IEEE Workhop on Applications of Computer Vision, 1992, pp. 11–18.

[13] D. Koller, K. Daniilidis, and H. H. Nagel: Model-nased object tracking in monocular omage sequences of road traffic scenes. Internat. J. Computer Vision *10* (1993), 257–281.

[14] A. Krause, J. Leskovec, and C. Guestrin: Data association for topic intensity tracking. In: Proc. 23rd International Conference on Machine Learning ICML'06, ACM Press, 2006, pp. 497–504.

[15] S. Kyo, T. Koga, K. Sakurai, and Shin'ichiro Okazaki: A robust vehicle detecting and tracking system for wet weather conditions using the IMAP-vision image processing board. In: Proc. Intelligent Transportation Systems, IEEE *99*, 1999, pp. 423–428.

[16] B. Lee and E. Cha: Fast and robust techniques for detection of car plate using HSV and weighted morphology. http://164.125.165.168/˜lbmo/html/upload_file, 2002.

[17] B. L. Lim, W. Yeo, K. T. Tan, and C. Y. Teo: A Novel DSP based real-time character classification and recognition algorithm for car plate detection and recognition. In: Proc. ICSP '98 Fourth International Conference on Signal Processing IEEE, Beijing 1998, pp. 1269–1272.

[18] Ch.-P. Lin, J.-Ch. Tai, and K.-T. Song: Traffic monitoring based on real-time image tracking. In: Proc. Robotics and Automation, IEEE *03* (2003), pp. 2091–2096.

[19] F. Martin and D. Borges: Automatic car plate recognition using partial segmentation algorithm. In: Proc. Signal Processing, Pattern Recognition, and Applications 2003, Rhodes, 404-061.

[20] Z. Michele, M. Stefano, and M. M. Carla: An efficient vehicle queue detection system based on image processing. In: Proc. 12th International Conference on Image Analysis and Processing (ICIAP'03), IEEE *03* (2003), pp. 232–237.

[21] F. A. Mujica, J.-P. Leduc, R. Murenzi, and M. J. T. Smith: A new motion parameter estimation algorithm based on the continuous wavelet transform. IEEE Trans. Image Processing (2000), 873–888.

[22] Z. B. Musa and J. Watada: A grid-computing based multi-camera tracking system for vehicle plate recognition. In: Proc. Czech–Japan Seminar, Kitakyushu 2006, pp. 184–189.

[23] Z. Naor: Tracking mobile users with uncertain parameters. Wireless Networks *9* (2003), 6, 637–646.

[24] I. Pitas: Digital Image Processing Algorithms. (Prentice Hall International Series in Acoustics, Speech and Signal Processing.) Prentice Hall, Englewood Cliffs, N.J. 1993.

[25] A. Prati, R. Vezzani, L. Benini, E. Farella, and P. Zappi: Enlarge and enhance the view with video, audio and sensor networks: An integrated multi-modal sensor network for video surveillance. In: Proc. Third ACM International Workshop on Video Surveillance & Sensor Networks VSSN'05, ACM Press 2005, pp. 95–102.

[26] K. K. Sang, W. K. Dae, and J. K. Hang: A recognition of vehicle license plate using genetic algorithm based segmentation. In: Proc. Internat. Conference on Image Processing 1996, pp. 661–664.

[27] T. Sato: Technical view: Situation recognition and its future in ubiquitous society – human support systems in terms of environmental system and contents system. In: Special Issue on Situation/Context Awareness Technologies for Human Support (T. Sato, ed.), J. Systems Control Inform. *49* (2005), 4.
[28] Y. Seto: Trend of biometric security technology. In: Special Issue: Advances in Biometric Identification (Yoichi Seto, ed.), J. Soc. Instrum. Control Engrg. *43* (2004), 7.
[29] R. Stefano and Z. Rodolfo: A multiprocessor-oriented visual tracking system. IEEE Trans. Industrial Electronic *46* (1999), 4, 842–850.
[30] S. Uchihashi: Video applications: Improvising camera control for capturing meeting activities using a floor plan. In: Proc. Ninth ACM International Conference on Multimedia, ACM Press, 2001, pp. 12–18.
[31] Y.-F. Wang, E. Y. Chang, and K. P. Cheng: Enlarge and enhance the view with video, audio and sensor networks: A video analysis framework for soft biometry security surveillance. In: Proc. Third ACM International Workshop on Video Surveillance & Sensor Networks VSSN'05, ACM Press, 2005.
[32] J. Watada and Z. B. Musa: A future view of a multi-camera tracking system. In: Proc. SICE-ICCAS 2006, Organized Session: SICE City, Busan 2006, pp. 71–78.

Zalili Binti Musa and Junzo Watada, Graduate School of Information, Production and Systems, Waseda University, 2-7 Hibikino, Wakamatsu, Kitakyushu 808-0135. Japan.
e-mails: zalili@kuktem.edu.my, junzow@osb.att.ne.jp