Toward Real-time Vehicle Detection Using Stereo Vision and an Evolutionary Algorithm

Vinh Dinh Nguyen, Thuy Tuong Nguyen, Dung Duc Nguyen, and Jae Wook Jeon, *Member, IEEE*

***Abstract*— A new approach for vehicle detection and distance estimation based on stereo vision and evolutionary algorithm (SEA) is described in this paper. First, we reuse our recent work on FPGA implementation of census-based correlations for stereo matching. Next, the SEA uses the gray scale left image and disparity information obtained from the FPGA system to detect the preceding vehicle and estimate its distance. This paper introduces an effective fitness function that allows our proposed method to have an improved performance and higher accuracy when compared with the existing evolutionary algorithm (EA) based methods. A new crossover type, tournament crossover, is introduced to reduce the convergence time of our proposed. This paper also introduces a new approach for estimating the fitness function parameters. This estimation differs from the traditional EA because these parameters were generally created via experiments. Moreover, the processing time and accuracy of SEA can be improved by converting the global search to the local search with V disparity map. The robust experiments have proved that SEA successfully detects vehicles in front and sustains noise from different objects appearing along the road. The detection range is 10m-140m, the detection rate is 95 % and the average processing-time is approximately 31 ms/frame on CPU. These results prove that SEA is suitable for a real-time system.**

I. INTRODUCTION

Every hour, thousands of people worldwide die or become disabled due to vehicle accidents. Therefore, many advanced driver assistance systems and traffic surveillance systems [1] have been built by numerous researchers to support drivers and reduce accidents. Many methods have been proposed to build these systems in computer vision. These methods can be classified into the following three main categories: knowledge-based methods, stereo-vision-based methods, and motion-based methods, as described in [2]. This paper introduces a brief discussion of vehicle detection methods by classifying them into smaller categories. This work also analyzes the strengths and weaknesses of each category.

Feature-Based Method: Feature-based methods use vehicles features to determine where the vehicle is located in the image. These features could be the color, shadow, texture, symmetry or lights of the vehicle, as in [3]. However, it is difficult to identify a distinct feature distinguishing between a non-vehicle and a vehicle.

Stereo Vision-Based Method: Disparity map and inverse perspective mapping [4] are two technique types used to ascertain stereo information for detecting a vehicle. The algorithm [4] symmetrically moves the left image to the right and the right image to the left to calculate the minimum non-similarity of each cell. The main problem of this algorithm is that too much time is consumed in moving and calculating the non-similarity.

Learning-Based Method: Two databases are used to detect a preceding vehicle: positive and negative. Vehicle images are stored in the positive database and non-vehicle images are stored in the negative database. These databases are then used by different learning algorithms [5]. The accuracy of learning-based methods is very high. However, the algorithm is designed to detect a specific vehicle and it cannot be used to detect other vehicle types.

Motion-Based Method: Anything that moves on the road is assumed to be a vehicle in the motion-based method. Therefore, several systems proposed in [6] use optical flow to detect a preceding vehicle. However, if there is no motion corresponding to a preceding vehicle, then this method cannot detect these vehicles.

Evolutionary Algorithm-Based Method: The common approach of a genetic algorithm for vehicle detection involves randomly generating all possible chromosomes, where each chromosome represents a feasible solution. Then, all chromosomes evolve via natural selection and genetics until a final solution is obtained. Two methods based on this approach are proposed in [7] [8]. These methods can detect the obstacle on the road but they cannot sustain noises coming from the pavement, buildings or crowd.

Many vehicle detection methods have been proposed based on five methods above, but there are a few approaches for vehicle detection based on the evolutionary algorithm. The processing time and accuracy of the existing EA [7][8] is not appropriate for real-time systems. Therefore, a new approach for vehicle detection based on SEA is proposed within this paper. Our proposed method is faster and more accurate than existing EA methods for vehicle detection. Moreover, our proposed method also achieves real-time processing and high accuracy when compared to the four remaining approaches. SEA uses the edge information [9] on the left image and the disparity information to detect a preceding vehicle. Then, the detected vehicle is tracked by reducing the search space and population size. Moreover, this paper also suggests a new approach to determine the fitness function parameters using EA and the least squares method. In addition, the preceding vehicle's location is the most dangerous when it occurs in our

This research was performed as part of the Hyundai Kia Motors project and Priority Research Centers Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(2011-0018397).

V. D. Nguyen, T. T. Nguyen, D. D. Nguyen and J. W. Jeon are with School of Information and Communication Engineering, Sungkyunkwan University, Korea (email: vinhnd@skku.edu, ntthuy@skku.edu, nddunga3@skku.edu, jwjeon@yurim.skku.ac.kr).

978-1-4673-0990-5/12/$31.00 ©2012 IEEE

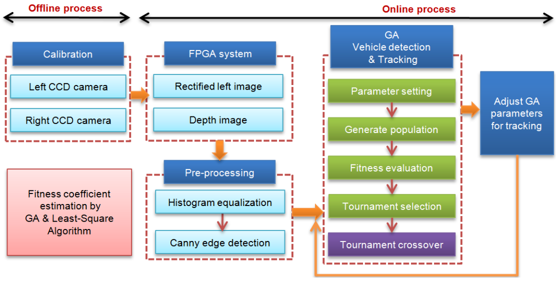


Fig. 1. The Flowchart of our proposed system.

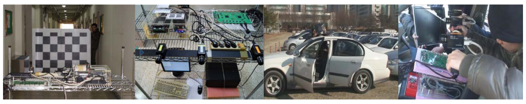


Fig. 2. Practical model of our proposed system.

car lane. Therefore, our proposed method detects the nearest vehicle rather than other positions.

The remainder of this paper is organized as follows. Details of our proposed algorithm are introduced in Section II. Experimental results and comparison of our algorithm to other are presented in Section III. Finally, conclusion of this paper is presented in Section IV.

II. Vehicle Detection and Distance Estimation Based on SEA

The two main processes used to describe SEA are offline and online processes. The offline process consists of performing stereo camera calibration as well as estimating of the weighting constants of the fitness function. The online process compromises four main stages: disparity calculation, pre-processing [9][10], vehicle detection and tracking phase. Fig. 1 depicts the block diagram of our SEA.

A. Stereo Camera Calibration and Stereo Matching

This paper uses [11] for calibration. We used a large chessboard 2m x 1.5m (width and height) to calibrate a far distance as well as to obtain a high quality disparity map as shown in Fig. 2. We used a 16mm and a 25mm lens for calibration. The baseline used for the calibration was 35cm and 60cm. Our proposed method used different baselines and focal lengths to determine the possible detection range. Computing the disparity map is very time consuming due to find the correspondence of each pixel between the left and right images on the same row. However, it is possible to run this in real-time using our previous work on the FPGA [12].

B. Vehicle Detection Based on SEA

1) Chromosome Structure: The selection of an appropriate genotype representation is important work for enhancing the processing time for an evolutionary algorithm. Three genotype representations commonly used in the context of genetic algorithms and genetic programming are string, matrix and tree structures. This paper proposes a chromosome structure based upon the string representation. Each chromosome represents a vehicle candidate, as shown in Fig. 3. The phenotypic trait of a chromosome is a rectangular shape. Each chromosome is organized by four genes. $G_{ux}^{(i)}$ and $G_{uy}^{(i)}$ represent the x-coordinate and the y-coordinate of the ith chromosome's upper-left corner, respectively. $G_{lx}^{(i)}$ and $G_{ly}^{(i)}$ represent the x-coordinate and y-coordinate of the ith chromosome's lower-right corner, respectively.

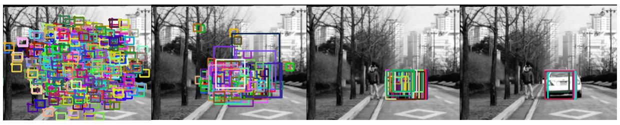


Fig. 3. Left to right: initial population, first, third and fifth generations.

2) Initial Population: Population of chromosomes representing vehicle candidates is randomly initialized within the region of the left image as

$$\begin{aligned} x_i^c &= x_i + w_{i,x}^c \\ y_i^c &= y_i + w_{i,y}^c \end{aligned} \tag{1}$$

where (x_i^c, y_i^c) is the position of the upper left corner of chromosome at time index i. (x_i, y_i) is the position of the image's center,and $w_{i,x}^c$ and $w_{i,y}^c$ are zero mean Gaussian processes with standard deviations of σ_x^c and σ_y^c, respectively. The phenotype of each random chromosome is a rectangular shape. Width and height of the rectangle are 20 pixels and 15 pixels, respectively. Fig. 3 shows the population after generating the chromosomes based on the Gaussian process.

3) Fitness Evaluation and Tournament Selection: The fitness of an individual in a genetic algorithm is the value of an objective function for its phenotype. The fitness function not only indicates how good the solution is, but also corresponds to how close the chromosome is to being optimal one. This paper proposes an effective fitness function to evaluate the fitness of each chromosome.

$$\begin{aligned} F^{(i)} = &- C_{de}G_{de}^{(i)} + C_l G_l^{(i)} + C_r G_r^{(i)} + C_t G_t^{(i)} + C_b G_b^{(i)} \\ &+ C_d G_d^{(i)} + C_{rd} G_{rd}^{(i)} \end{aligned} \tag{2}$$

$$G_{de}^{(i)} = \frac{1}{w^{(i)}h^{(i)}} \left(w^{(i)}h^{(i)} - \sum_{j=0}^{h^{(i)}} \sum_{k=0}^{w^{(i)}} d(j,k) \right) \tag{3}$$

$$G_l^{(i)} = \frac{1}{3h^{(i)}} \sum_{j=0}^{h^{(i)}} \sum_{k=-1}^{1} I_{egde}(G_{uy}^{(i)} + j, G_{ux}^{(i)} + k) \tag{4}$$

$$G_r^{(i)} = \frac{1}{3h^{(i)}} \sum_{j=0}^{h^{(i)}} \sum_{k=-1}^{1} I_{egde}(G_{uy}^{(i)} + j, G_{lx}^{(i)} + k) \tag{5}$$

$$G_t^{(i)} = \frac{1}{3w^{(i)}} \sum_{j=-1}^{1} \sum_{k=0}^{w^{(i)}} I_{egde}(G_{uy}^{(i)} + j, G_{ux}^{(i)} + k) \tag{6}$$

$$G_b^{(i)} = \frac{1}{3w^{(i)}} \sum_{j=-1}^{1} \sum_{k=0}^{w^{(i)}} I_{egde}(G_{ly}^{(i)} + j, G_{ux}^{(i)} + k) \tag{7}$$

$$G_d^{(i)} = \frac{1}{\lambda_{\max} D_v} fb \tag{8}$$

$$G_{rd}^{(i)} = 1 - \frac{\sqrt{\left(G_{ux}^{(i)} - x_{center}\right)^2 + \left(G_{uy}^{(i)} - y_{center}\right)^2}}{\sqrt{\left(-\frac{1}{2}G_{ux}^{(i)}\right)^2 + \frac{1}{4}\left(G_{uy}^{(i)}\right)^2}} \tag{9}$$

$$d(i,j) = \left\{ \begin{array}{l} 1,\ d_v(i,j) = D_v \\ 0,\ d_v(i,j) \neq D_v \end{array} \right.$$

$$C_{de} + C_l + C_r + C_t + C_b + C_d + C_{rd} = 1 \tag{10}$$

where (2) represents the fitness value of the ith chromosome and (10) represents weighting positive constants. These constants can be approximated by using the least squares method and genetic algorithm. (3) represents the disparity error of the ith chromosome by mapping its position in relation to the disparity image and then calculating the number of pixels with intensities that are not equal to the dominant disparity value $D_v^{(i)}$. (4) and (5) represent the density of the edge pixels along the left side and the right side of the ith chromosome, respectively. (6) and (7) represent the density of the edge pixels along the top side and the bottom side of the ith chromosome, respectively. (8) is the ratio representing the distance from the camera system to the ith chromosome; where f is the focal length(in pixels), b is the baseline(in meters), and $\lambda_{\max}$ is the maximum detected distance of our proposed method ($\lambda_{\max} = 140m$). (8) is designed to detect the nearest vehicle from the camera system and avoid the missing detections which come from far buildings. (9) represents the ith chromosome's position with regard to the image's center. $d_v(i,j)$ is the intensity value of pixel (i,j) in the disparity map. $w^{(i)}$ and $h^{(i)}$ represent the width and height of the chromosome in pixels. $I_{egde}(i,j)$ is the intensity value at pixel (i,j) in the edge image. (x_{center}, y_{center}) is the coordinate of image's center. $D_v^{(i)}$ is the intensity value represented to the disparity of ith chromosome. This disparity is calculated by mapping the ith chromosome into the disparity map and finding the pixel intensity value with the maximum density. (9) is built to ensure that the fitness value of an object located near the image center is higher than other objects appearing along the pavement, which may cause false detections. The vehicle located near the image center is more dangerous when compared to other positions. Thus, we endeavored to design a function that increases the fitness value of the vehicle located near the image center and decreases the fitness value of other positions. This work also helps us remove false detections. If the vehicle is not located near the image center, it can also be detected due to the following three reasons: 1) If the non-vehicle is located near the image center, it is dangerous to the driver and, therefore it should be detected. 2) If there is no object located near the image center, then the fitness value of the vehicle will be higher in comparison to other non-vehicle. and, 3) After the detecting phase, the tracking phase starts to track the preceding vehicle, so the population will be generated inside the region tracking. The next population is established by improved chromosome's quality from current population. The selection method used in our algorithm is a ten-tournament selection proposed in [13].

4) Tournament Crossover: Many crossover types are used to produce a new child in EA, such as n-point crossover or uniform crossover. This paper introduces a new crossover type, tournament crossover, which leads our method to have an earlier convergence. Two chromosomes, an ith and a jth, are randomly selected from the selection population, and then the kth child is produced by considering the disparity and coordinates of the two parents. If the two parents have similar disparity, then the following condition will be executed to produce two new children with the same feature. Otherwise, two parents are directly moved to the next generation. Thus, the new child's area will cover the entire area of two parents. The coordinate of new chromosome is calculated as follows:

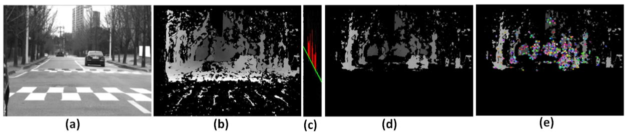


Fig. 4. Limiting the search space using V disparity map.

$$\begin{array}{l} G_{ux}^{child} = \min(G_{ux}^{(i)}, G_{ux}^{(j)}), G_{uy}^{child} = \min(G_{uy}^{(i)}, G_{uy}^{(j)}) \\ G_{lx}^{child} = \max(G_{lx}^{(i)}, G_{lx}^{(j)}), G_{ly}^{child} = \max(G_{ly}^{(i)}, G_{ly}^{(j)}), \end{array} \tag{11}$$

The similar disparity between two chromosomes are determined by using Census transform on the dominant disparity value of each chromosome. Then, the cost of two Census transformed pixels are calculated using Hamming distance:

$$C_{\cos t} = Hamming(C_1(i_1, j_1), C_2(i_2, j_2)) \tag{12}$$

where $C_1(i_1, j_1)$ is the bit strings of the dominant pixel (i_1, j_1) of the first chromosome. $C_2(i_2, j_2)$ is the bit strings of the dominant pixel (i_2, j_2) of the second chromosome.

In order to avoid false detections, a new chromosome needs to be validated before moving it to the next generation by checking three following properties: 1) The chromosome's edge information must satisfy the U-shape. 2) the width and height the chromosome must correspond. 3) the chromosomes width (in pixels) and the disparity must correspond. The relationship between the vehicle's width and its disparity is described as $W_p = \frac{W_m}{b} d$, where W_p is the vehicle width(in pixels); W_m is the vehicle width (in meters).

C. Vehicle Tracking-based Genetic Algorithm

Once the preceding vehicle is detected, it can be tracked based on the original left image. We also use the same algorithm for detecting phase, and the population size and searching space are reduced based on previous detection results. Reducing the population size and the searching space can reduce the processing time. The searching space (x, y) in the current left image is determined based on the position of the previous detection.

$$\begin{array}{c} x_u^l - w \leq x \leq x_{lo}^r + w,\ w = x_{lo}^r - x_u^l \\ y_u^l - h \leq y \leq y_{lo}^r + h,\ h = y_{lo}^r - y_u^l \end{array} \tag{13}$$

where (x_u^l, y_u^l) and (x_{lo}^r, y_{lo}^r) are the position of the vehicle's upper-left corner and the vehicle's lower-right corner in the previous detection. The population size is reduced by half from the detection phase.

D. GA Parameters Estimation

The weighting positive constants of the fitness function are estimated using both genetic algorithm and the least square method. If we already know the position of the chromosome representing the vehicle in the left image, then we try to find the weighting constants that make the fitness value of the vehicle-chromosome is highest among other non-vehicle-chromosomes. Of course, the vehicle can appear at any place in the image, so this estimation must cover all possible vehicles positions, which means that we have to perform this estimation on a set of images. Supposing that, we randomly generate N individual in the image, and then $F^{(i)}$ is the fitness function of the ith chromosome, and the k^{th} chromosome is the optimal solution representing vehicle:

$$\begin{aligned} F^{(i)} = & -C_{de}G_{de}^{(i)} + C_l G_l^{(i)} + C_r G_r^{(i)} + C_t G_t^{(i)} + C_b G_b^{(i)} \\ & + C_d G_d^{(i)} + C_{rd} G_{rd}^{(i)};\ i \in [1, ..., k, ..., N] \end{aligned} \tag{14}$$

We calculate the weighting constants for each pair $\{F^{(k)}, F^{(i)}\}; i \neq k$ by:

$$\max\ F^{(k)} - F^{(i)},\ subject\ to \begin{cases} \sum\limits_{\Omega} C_{\{x\}} \leq 1; C_{\{x, x \in \Omega\}} > 0 \\ F^{(k)} > F^{(i)} \\ \Omega = \{C_{de}, C_l, C_t, C_b, C_d, C_{rd}\} \end{cases}$$

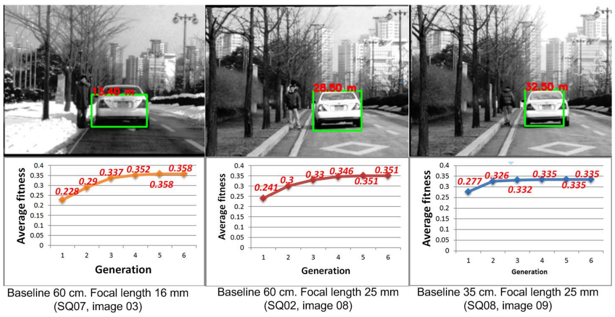
Next, least square method [14] is applied to find the weighting constants from the result of each pair $\{F^{(k)}, F^{(i)}\}$. We have used 115 images to estimate the weighting constants of the fitness function. The vehicle is located at different positions in the image so that we can cover all possible solutions. We obtain $C_l = 0.17$, $C_r = 0.05$, $C_t = 0.25$, $C_b = 0.24$, $C_{de} = 0.05$, $C_d = 0.05$ and $C_{rd} = 0.18$ after performing it upon 115 images. These constants are then used to run our proposed algorithm with all the test images.

E. Limiting the searching space of SEA using V-disparity

The search space is very important in SEA. It decides the quality and convergence time to obtain the optimal solution. In previous discussion, all possible solutions are generated in the entire of the image's region (global search), so much time is consumed to process the non-obstacle regions such as road. This problem is the main disadvantage of the proposed methods in [7] [8]. However, these issues can be solved by only generating the chromosomes on the surface of the obstacles based on the V disparity map in [15] (local search). The blue line and the red line in the Fig.4 (c) show the road and obstacles, respectively. These lines can be detected by using Hough Transform in [15]]. We only need generating the chromosomes on the obstacle's surfaces. Fig.4 (d) and Fig.4 (e) show the search space and chromosomes after using V disparity map, respectively. Thus, the convergence time and accuracy of our proposed method enhanced by converting the search space from global to local.

F. Convergence Time

Convergence time is the number of generations that perform until the population converges. This means that all chromosomes in the population are the same or the fitness value of every chromosome in the population is equal. In the crossover stage, two chromosomes combine together to create a new child. There are many chromosomes located inside the vehicle region, so more than one pair of chromosomes can produce the same child. Therefore, the convergence time of our proposed algorithm is faster than the standard GA. Moreover, the convergence time will be reduced after the detecting phase due to the start of the tracking phase. Thus, there are two main reasons for an earlier convergence comparing to other GAs: (1) The searching space is limited from the tracking phase and V disparity information; (2) More than one pair of chromosomes can produce the same child due to their locations. Fig. 3 shows the detected result after five generations. Fig. 5 shows three test images with the average fitness and generation.



Baseline 60 cm. Focal length 16 mm (SQ07, image 03)

Baseline 60 cm. Focal length 25 mm (SQ02, image 08)

Baseline 35 cm. Focal length 25 mm (SQ08, image 09)

Fig. 5. The flow chart represents the average fitness for each generation.

III. Experimental Results

Our proposed algorithm implemented in a CPU with an AMD Athlon x2 CPU 2.9GHz. The stereo vision system on a FPGA [12] was implemented using a Virtex-4 XC4VLX200-10 FPGA from Xilinx. Two CCD cameras with four different stereo configurations, $[b = 35cm, f = 16mm]$, $[b = 35cm, f = 25mm]$, $[b = 60cm, f = 16mm]$, and $[b = 60cm, f = 25mm]$, are used to capture the left and right images of a preceding vehicle. Gray-scale left and disparity images, at a resolution of 320 x 240 (pixels) outputted from the FPGA system, are used to run vehicle detection on the CPU. Ten sequences with total 2265 images were used in the experiment. The population size and the number of generations used in our proposed algorithm are 1000 and 5, respectively. Fig. 6 shows vehicle detection and distance estimation under different conditions after using histogram equalization[10].

We tested our proposed method in two cases. First, we move our system into a portable car and capture the image on highway from our university to Bucheon technopark. Fig. 6(a) and 6(b) shows detection results in difficult situations, such as numerous buildings, trees and other obstacles appearing in the real condition. Fig. 6(c) and 6 (d), are images captured under the snow and many trees. Fig. 6(e) and 6(f), are images captured under normal conditions. Second, we move the system to the car simulator system and capture images on a university's road as in Fig. 6(g) and 6(h). For 2265 experimental images, our proposed system successfully detects and then estimates

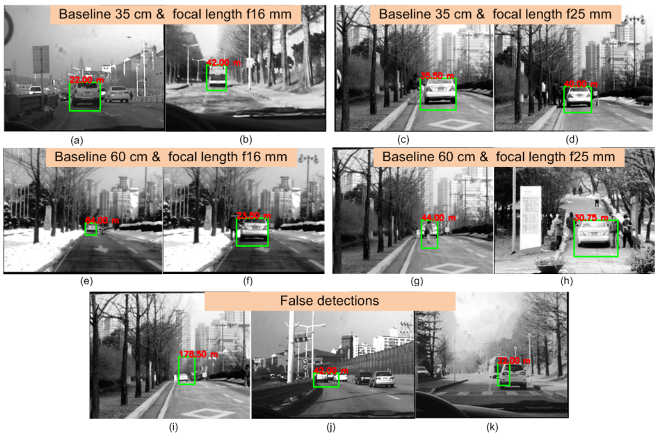

Baseline 35 cm & focal length f16 mm
22.00 m
42.00 m
(a)
(b)
Baseline 35 cm & focal length f25 mm
35.50 m
40.00 m
(c)
(d)
Baseline 60 cm & focal length f16 mm
64.00 m
23.50 m
(e)
(f)
Baseline 60 cm & focal length f25 mm
44.00 m
30.75 m
(g)
(h)
False detections
178.50 m
42.00 m
22.00 m
(i)
(j)
(k)

Fig. 6. Experimental results under various road conditions.

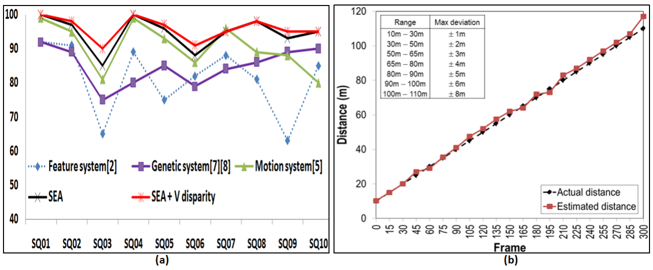


Fig. 7. (a) Average detection rate (DR) of our proposed algorithm in comparison with other systems; (b) Average of estimated distances compared to the actual distances.

the distance of the vehicles in front. Detection ranges for $[b = 35cm, f = 16mm]$, $[b = 35cm, f = 25mm]$, $[b = 60cm, f = 16mm]$ and $[b = 60cm, f = 25mm]$ are $10m - 60m$, $25m - 130m$, $10m - 70m$ and $30m - 140m$, respectively. We implemented three existing methods, the motion-based method M [6], the feature F [3] [16] based-method and the genetic based-method G [7][8], to compare them with our proposed method. In M, motion changes can be effectively detected using the difference between two successive images. The interframe difference (IFD) is determined by performing a subtraction between two images based on pixels. F detects a vehicle by performing edge detection, and then using a pre-defined pattern to detect the upper part of the preceding vehicle. Finally, the detected region is expanded to the entire vehicle based on the disparity information. G detects a vehicle by performing edge matching between the left and the right images and the best match is considered to be the vehicle candidate. Fig. 7(a) shows the average detection rate(DR) comparison of our proposed method to these existing methods on ten image sequences. DR is calculated by detecting the nearest vehicle in our lane. Some instances of false detections occurred due to very high illumination, many vehicles located next to each other, or a low quality of disparity due to the weather conditions. Fig. 6(f), 6(g) and 6(k) show the detected results for the false detection cases. Fig. 6(g) shows the detected results for the occlusion cases. Processing time of M, F, G, SEA (without V disparity) and SEA+ (with V disparity) is 11ms, 33ms, 55ms, 45ms and 31ms, respectively. In DR comparison, our proposed method is better than M, F and G. In processing time comparison, our proposed method is better than F, G and less than M. Fig. 7(b) presents the average estimated distances compared to the actual distance in frame indices. In addition, maximum errors corresponding to distance ranges were shown in this figure.

IV. Conclusion

This paper proposes and discusses a new approach for real-time vehicle detection and tracking using both stereo vision and genetic algorithm methods. The robust experiments under different scenarios, including intersections, oncoming traffic, occlusions, and crowds on both country and highway roads have proven that the proposed system is able to accurately detect and estimate the distance of the vehicles in front. We plan to port our system to a NVIDIA GeForce GTX 460 to achieve high applicability in a vehicle computing platform and enhance the proposed method to detect multiple vehicles at the same time.

References

[1] T. T. Nguyen, X. D. Pham, J. H. Song, S. Jin, D. Kim, J. W. Jeon, "Compensating Background for Noise due to Camera Vibration in Uncalibrated-Camera-Based Vehicle Speed Measurement System," *IEEE Trans. Veh. Technol.*, 60(1):30-43, 2010.

[2] Z. Sun, B. George, and M. Ronald, "On-Road Vehicle detection: A Review," *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(5):694-771, 2006.

[3] M. Bertozzi, A. Broggi, and S. Castelluccio, "A real-time oriented system for vehicle detection," *J. of Syst. Archit.*, pp. 317-325, 1997.

[4] Y. Xu, M. Zhao, X. Wang, Y. Zhang, Y. Peng, Y. Yuan, H. Liu, "A Method of Stereo Obstacle Detection Based on Image Symmetrical Move," *in Proc. IEEE Symp. Intell. Veh.* , pp. 36-41, 2009.

[5] Fan Jiang, Xinggang Lin, "A Learning Based Approach for Vehicle Detection," *IEEE conf. on TENCON*, pp. 1-4, 2006.

[6] T. Naito, T. Ito, Y. Kaneda, "The Obstacle Detection Method using Optical Flow Estimation at the Edge Image," *in Proc. IEEE Symp. Intell. Veh.*, pp. 817-822, 2007.

[7] O. Pauplin, J. Louchet, E. Lutton and A. L. Fortelle, "Evolutionary Optimisation for Obstacle Detection and Avoidance in Mobile Robotics," *J. of Advanced Comput. Intell. Intell. Inform.*, 9(6):622-629, 2005.

[8] Y. Ruicheck, H. Issa, J.G. Postaire, J.C. Burie, "Towards Real-Time Obstacle Detection Using a Hierarchical Decomposition Methodology for Stereo Matching with a Genetic Algorithm," *IEEE Int. Conf. Tools with Artif. Intell.*, pp.138-147, 2004.

[9] J. Canny , "A Computational Approach To Edge Detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp.679-698, 1986.

[10] Bernd Jhne, "Digital Image Processing," 6nd edition, *Springer*, pp. 91-142, 2005.

[11] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(11):1330-1334, 2000.

[12] S. Jin, J. U. Cho, X. D. Pham, K. M. Lee, S. K. Park, M. Kim and J. W. Jeon, "FPGA Design and Implementation of a Real-Time Stereo Vision System," *IEEE Trans. Circuits Syst. Video Technol.*, 20(1):15-26, 2010.

[13] D.E. Goldberg, "Genetic Algorithms in Search," *Addison-Wesley, Reading*, MA (1989).

[14] Edwin K. P. Chong and Stanislaw H. Zak, "An Introduction to Optimization, 2nd Edition," 2nd edition, *JOHN WILEY & SONS, INC*, pp. 187-212, 2001.

[15] R. Labayrade, D. Aubert D, J. P. Tarel, "Real Time Obstacle Detection on Non Flat Road Geometry through V-Disparity Representation," *in Proc. IEEE Symp. Intell. Veh.*, pp. 2(1):646-651, 2002.

[16] V. D. Nguyen, T. T. Nguyen, D. D. Nguyen, J. W. Jeon, " Real-time vehicle detection design and implementation on GPU," *in Proc. IEEE Int. Conf. ICASS*, pp.1287-1292, 2011.