An FPGA-based Collision Warning System Using Hybrid Approach

Haichao Liang, Takashi Morie, Youhei Suzuki, Kazuki Nakada, Tsutomu Miki, and Hatsuo Hayashi
Kyushu Institute of Technology
Graduate School of Life Science and Systems Engineering
2-4, Hibikino, Wakamatsu-ku, Kitakyushu, 808-0196 Japan
liang-haichao@edu.brain.kyutech.ac.jp

Abstract

In this paper, we propose an FPGA-based collision warning system for advanced automobile driver assistance systems or autonomous moving robots. The system consists of three function blocks: coarse edge detection using a resistive-fuse network, moving-object detection inspired by neuronal propagation in the hippocampus, and danger evaluation and collision warning using fuzzy inference. The first two functions are implemented in FPGAs. The system can detect moving objects with a speed range of 3-192 km/h with a sampling period of 30 ms for an input image of 320 × 256 pixels, and can output a warning against dangerous regions in the input image.

1 Introduction

The advanced driver assistance systems (ADAS) attracts growing interest to decrease road traffic accidents. In the ADAS, the forward vehicle collision warning system (FVCWS) is useful to decrease impact accidents which is the most important reason causes injury or death of people in road traffic accidents.

In order to realize the FVCWS, the most important processing is moving-object detection. So far, optical flow models using stereo camera systems [5, 6], or single-camera images [1, 7] are mainly used for that purpose. In stereo camera systems, the distance to vehicle is calculated by using the parallax and the principle of triangulation. A millimeter-wave radars are often used to improve the performance of detection accuracy and detection range [6]. The stereo camera systems provides high detection accuracy, and can calculate the optical flow, but calculation of parallax and velocity needs an image matching processing with high calculation costs. In the approaches using single-camera images, optical flow is calculated by using complicated mathematical models, which also requires a large amount of calculation costs.

Therefore, simple and high speed algorithms for moving object detection are needed. We have already proposed such an algorithm using single-camera images inspired by neuronal propagation in the hippocampus [3, 8], and have already implemented the algorithm in an FPGA [2].

In this paper, we propose an collision warning system based on the FPGA implementation of moving-object detection algorithm we proposed. Our system consists of three function blocks: coarse edge detection, moving-object detection, and danger evaluation and collision warning. In coarse edge detection, principal edges are only detected to reduce information complexity of the input image. We use a resistive-fuse network for that purpose, which has already been implemented in an FPGA [4]. Because the actual traffic patterns are very complicated, we use fuzzy inference to analyze the moving-object detection results in the danger evaluation processing.

Compared with conventional approaches, our system has an advantage of high speed operation and low calculation cost because it performs neither object recognition processing nor actual velocity calculation. Our system only treats object edges instead of objects themselves and evaluates the degree of danger of predefined regions. Even if an occlusion occurs, it hardly affects the evaluation results. Our system uses moving-detection results in each predefined direction instead of calculating the actual velocity, which also reduces the calculation cost. Although it needs a complicated fuzzy rules to evaluate the degree of danger, the total calculation cost of our system is much lower than that of conventional approaches. Therefore, we can develop a faster and more compact collision warning system.

2 FPGA-based Collision Warning System

2.1 Coarse Edge Detection

In this system, we use a resistive-fuse network to detect coarse edges [4]. An original resistive-fuse network circuit is composed of voltage sources, linear resistances and

0-7695-2946-1/07 $25.00 © 2007 IEEE
DOI 10.1109/HIS.2007.38

IEEE computer society

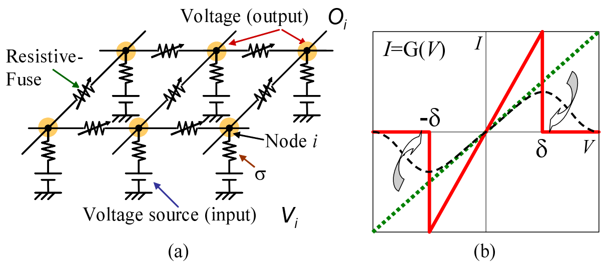

Figure 1. (a) Original analog resistive-fuse network circuit, (b) I-V characteristic of a resistive-fuse element.

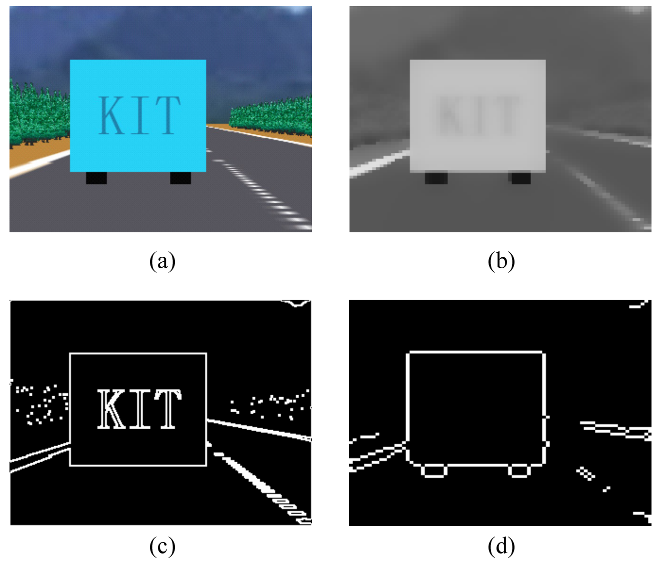

Figure 2. (a) Captured image, (b) resistive-fuse network processing result, (c) conventional edge extraction result, (d) edge data of the resistive-fuse network processing result.

resistive-fuse elements, as shown in Fig. 1(a). Each node corresponds to the pixel of the image. The value of each voltage source represents the image input V_i for pixel i. The voltage source and the node are connected via a linear resistance, whose conductance is σ. The adjacent nodes are connected with the resistive-fuse element. The I-V characteristic of this element is given by

$$I = G(V) = [\frac{1}{1+\exp\{\eta(V^2-\delta^2)\}}]\frac{V}{R_f} \qquad (1)$$

where η is a parameter for changing the conductance function, δ is a threshold value of the resistive-fuse element, and R_f is a resistance of the resistive-fuse element.

The conductance $G(V)$ is gradually changed from linear to the resistive-fuse characteristic by changing η, as shown in Fig. 1(b) to achieve the coarse region segmentation and the object contour extraction. At the first stage, the conductance is linear, the whole image is smoothed and averaged because of linear resistive network characteristics. At the final stage, the conductance becomes a resistive-fuse characteristic. This process is called *annealing*. If the voltage difference between the adjacent nodes is smaller than the threshold value ($|O_i - O_j| < \delta$), the region is smoothed and small regions disappear, because the network behaves as a simple linear resistive network. In contrast, if $|O_i - O_j| \geq \delta$, the pixel nodes are disconnected each other, and these pixels are considered as an image edge. Consequently, due to the annealing process, the contours of coarse regions are only detected as edges. Compared with the conventional approaches as shown in Fig. 2(c), the resistive-fuse network processing is useful to detect the object contours (Fig. 2(d)).

We have already implemented this algorithm in an FPGA [4]. We have designed a digital architecture with serial scanning and parallel neighboring-node interaction calculation. For each node, calculations with eight neighboring nodes are performed in parallel. The FPGA implementation also has an advantage that it can change the parameter easily.

Sometimes illumination changes could cause variation of edges. To avoid this problem, we can use some image processing, such as level adjustment and gamma correction before inputting the image data into the resistive-fuse network.

2.2 Moving-object Detection

The hippocampus is known as a brain area related with memory and spatial navigation in humans and animals. The hippocampal formation consists of three principal regions: the dentate gyrus (DG), CA3 and CA1. We proposed a moving-object detection algorithm inspired by neuronal propagation underlying sequence coding in a model of the hippocampus [8].

The proposed algorithm employs three kinds of maps: actual image (AC) map, CA3 map and CA1 map. The last two of them are named after the names of the hippocampus regions. They are divided into pieces based on radial coordinates and concentric rectangles as shown in Fig. 3. The CA3 map is divided into four submaps, and each CA3 submap is related to four corresponding CA1 submaps. A piece, which we call a unit, includes tens of pixels; the number of pixels in a unit, N, is large in the marginal regions of the maps, and is small in the central regions of the maps,

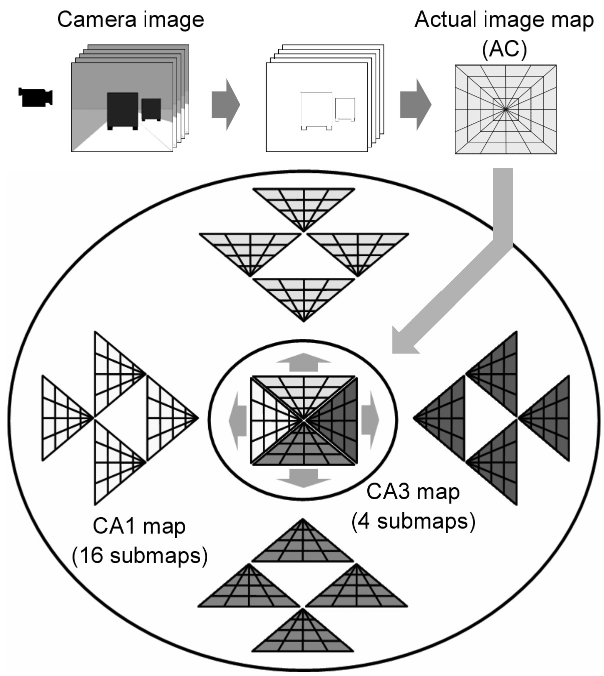

Figure 3. Moving object detection model inspired by neuronal propagation in hippocampus.

because the maps are divided radially. The range of N in this system is shown in Table 1.

The edge information of the input image is mapped onto the AC map. The AC map then generates unit-firing information which determines whether a CA3 unit fires or not according to the edge location in the corresponding AC unit. Each CA3 submap receives the unit-firing information and sends signals to four corresponding CA1 submaps that are specified to detect and represent objects moving in the direction of up, down, right and left by a pattern of activated units, respectively. Decaying of the value of each CA1 unit is used, instead of neuronal propagation, to measure how long it takes for each edge of moving objects to pass through a unit to the neighboring one, which is hereafter called *passing time*.

In the digital VLSI architecture design, we use a digital counter for the decaying signal generator to measure the passing time. In addition, the counter is shared with all units to reduce power dissipation. The detection principle of upward movement using our algorithm is shown in Fig. 4. If a CA3 unit receives a firing signal, it fires and holds the counter value at that time, and then sends the value to the neighboring CA1 units. Movement detection is accomplished by calculating the counter value difference between each CA3 unit and the corresponding CA1 unit in

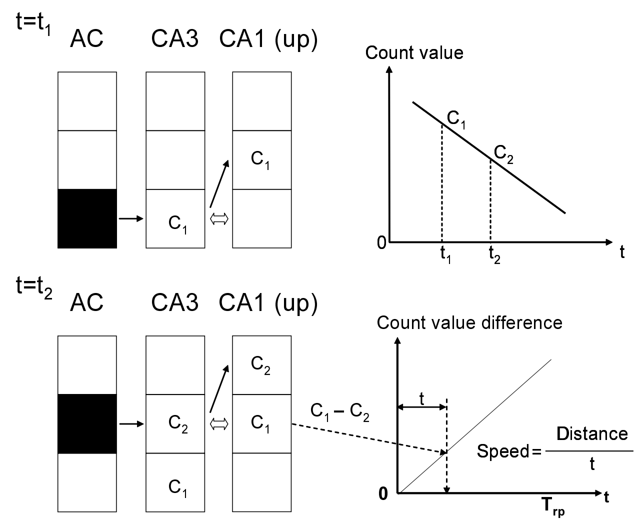

Figure 4. Detection of upward movement using our algorithm.

each detection direction. The passing time can be obtained from the counter value difference, which corresponds to the relative velocity because the distance between the neighboring units is fixed.

We can detect the relative speed of multiple moving objects in each predefined direction in high speed by using this simple algorithm. We have also implemented this algorithm in an FPGA [2], and the digital VLSI design result is shown in Fig. 5. The CA3-CA1 processor performs row-by-row parallel processing in each detection direction in each CA3 submap. We have verified its operation as shown in Fig. 6. The specification of the moving object detection part using an FPGA is shown in Table 1.

2.3 Danger Evaluation and Collision Warning Processing

In the danger evaluation, we use fuzzy inference to analyze the moving-object detection results and evaluate the degree of danger. A collision warning is output to a dangerous region. Although the CA3 map is divided into four submaps to detect moving objects as shown in Fig. 3, we additionally divide each CA3 submap into two parts in the radial direction to evaluate the degree of collision danger. We use a concentric rectangle for that purpose, whose height and width are half of those of the CA3 map, respectively. We call marginal regions *region-1* and central regions *region-2*, as shown in Fig. 7. Related to the divided CA3 submaps, the corresponding CA1 submap regions are defined.

We explain the danger evaluation processing by using

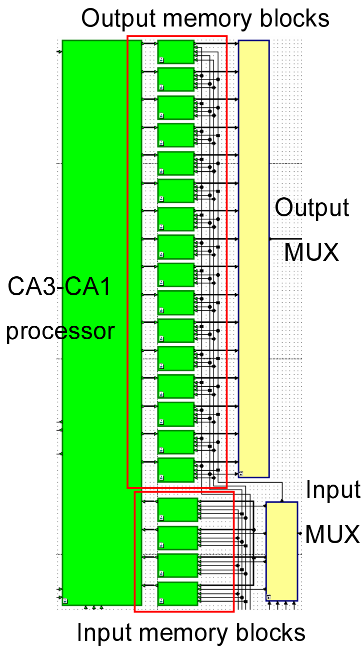

Figure 5. Block diagram of the moving-object detection processor.

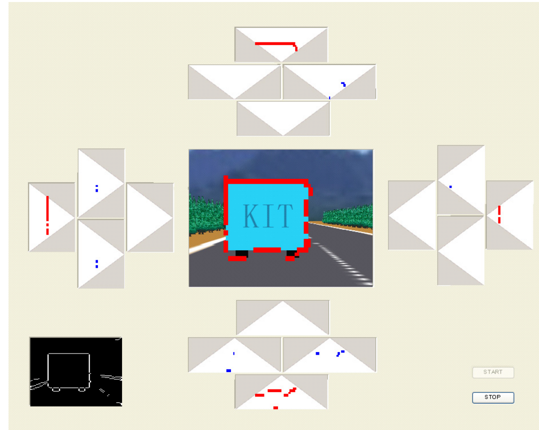

Figure 6. Approaching-object detection results using artificial simple moving images.

Table 1. Specification of moving object detection part

Image size (pixels)	320 × 256 (Width × Height)
Sampling period	30 ms
Number of units /submap	16 × 16
Number of pixels (N) /unit	48 – 288
CA1 decaying steps	64 steps (6-bit counter)
Refractory period	1,920 ms (30 ms × 64)
Passing time (unit to unit)	30 – 1,920 ms
Speed range of detection	3 – 192 Km/h
FPGA vendor	ALTERA
FPGA device	Stratix II (EP2S60F672C)
FPGA clock frequency	48 MHz
FPGA board	SX-USB II (Prime Systems,Inc)

car-to-car moving patterns shown in Fig. 8. For front impact, as shown in Fig. 8 (1) and (2), downward moving edges are detected in the bottom-region-1, and leftward and rightward moving edges are detected in the left-region-1 and the right-region-1, respectively. For front-to-side impact, as shown in Fig. 8 (3) and (4), downward moving edges are also detected in the bottom-region-1. In the cases shown in Fig. 8 (5) and (6), which are safe cases, downward moving edges are not detected in the bottom-region-1. Thus, we can define some if-then rules for evaluating the degree of collision danger using the detected moving edges according to the features described above.

We also use the average speed and acceleration of the detected moving edges to evaluate the degree of collision danger. A part of the defined if-then rules are as follows:

1. IF the number of downward moving edges detected in the (middle)-bottom-region-1 is *large/small*, THEN the degree of danger is *large/middle*.

2. IF the number of rightward moving edges detected in the right-region-2 is *large* AND their average speed is *high/low* AND downward moving edges are detected in the (right)-bottom-region-1, THEN the degree of danger is *large/middle*.

3. IF the number of rightward moving edges detected in the right-region-2 is *small* AND their average speed is *high/low* AND downward moving edges are detected in the (right)-bottom-region-1, THEN the degree of danger is *middle/small*.

4. IF rightward moving edges are detected in the right-region-2 AND downward moving edges are not detected in the (right)-bottom-region-1, THEN the degree of danger is *small*.

5. IF the number of rightward moving edges detected in the right-region-1 is *large* AND their average speed is *high* AND downward moving edges are detected in the (right)-bottom-region-1, THEN the degree of danger is *middle*.

6. IF the number of leftward moving edges detected in the right-region-1/2 is *large* AND their average ac-

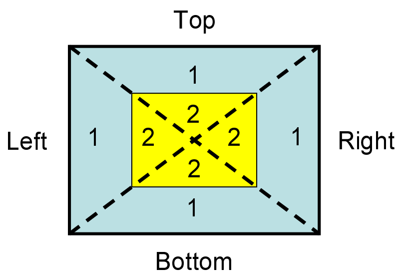

Figure 7. Map segmentation for danger evaluation.

celeration is *high* AND downward moving edges are detected in the (right)-bottom-region-1, THEN the degree of danger is *middle/large*.

7. IF the number of downward moving edges detected in the bottom-region-2 is *large* AND their average speed is *high*, THEN the degree of danger is *middle*.

Thus, using the coordinates, average speed and acceleration of the detected moving edges, we can evaluate the degree of collision danger based on fuzzy inference.

An example of fuzzy inference results is shown in Fig. 9. As described above, we use neither the UP-CA1 submaps nor upward CA1 submaps for evaluating the degree of danger. We only use 18 pieces of CA1 submaps for the danger evaluation, and we label them as shown in Fig. 9. The fuzzy inference result for each CA1 submap is expressed in the range from 0 to 1, and we set here the threshold value for danger evaluation as 0.5. In other words, if the fuzzy inference result is larger than 0.5, the system will output a warning using the predefined red color as shown in Fig. 9.

We verified the system operation using artificial simple moving images, and the results are shown in Fig. 9. It was verified that the system successfully output a warning against the approaching objects with red regions.

3 Conclusion

We proposed an FPGA-based collision warning system for the advanced driver assistance systems or autonomous moving robots. Using the fuzzy inference rules proposed for applying to the moving-edge detection patterns, we successfully verified the collision warning operation for artificial simple moving images. In the future work, we will apply our system to an autonomous moving robot to verify the system operation for real-world complicated moving images.

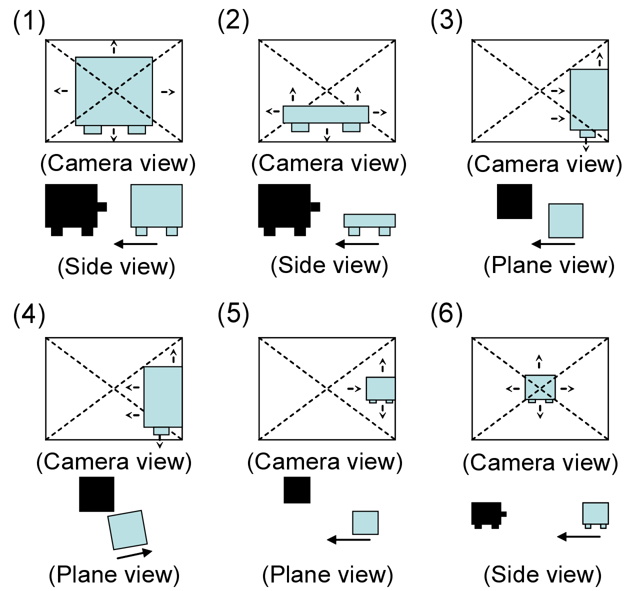

Figure 8. Car-to-car moving patterns for the collision warning system.

Acknowledgment

This work was partly supported by fund from MEXT, Japan, via a 21st Century COE program (center #J19) granted to Kyushu Institute of Technology.

References

[1] C. Brailon, C. Pradalier, J. L. Crowley, and C. Laugier. Real-time moving obstacle detection using optical flow models. In *IEEE Intelligent Vehicles Symposium*, pages 466–471, Tokyo, Japan, June 2006.

[2] H. Liang, T. Morie, H. Nakayama, K. Nakada, and H. Hayashi. An FPGA-based real-time moving-object detection system based on neuronal propagation in the hippocampus. In *RISP International Workshop on Nonlinear Circuits and Signal Processing (NCSP'07)*, pages 509–512, Shanghai, China, March 2007.

[3] H. Liang, H. Nakayama, K. Nakada, T. Morie, and H. Hayashi. Digital VLSI implementation of a moving object detection algorithm based on neuronal propagation in the hippocampus. In *Int. Symp. on Intell. Signal Processing and Comm. Systems (ISPACS2006)*, pages 614–617, Yonago, Japan, Dec. 2006.

[4] T. Nakano, T. Morie, H. Ishizu, H. Ando, and A. Iwata. FPGA implementation of resistive-fuse networks for coarse image-region segmentation. *Intelligent Automation and Soft Computing*, 12(3):307–316, 2006.

[5] S. Nedevsch, R. Danescu, D. Frentiu, T. Marita, F. Oniga, C. Pocol, R. Schmidt, and T. Graf. High accuracy stereo vi-

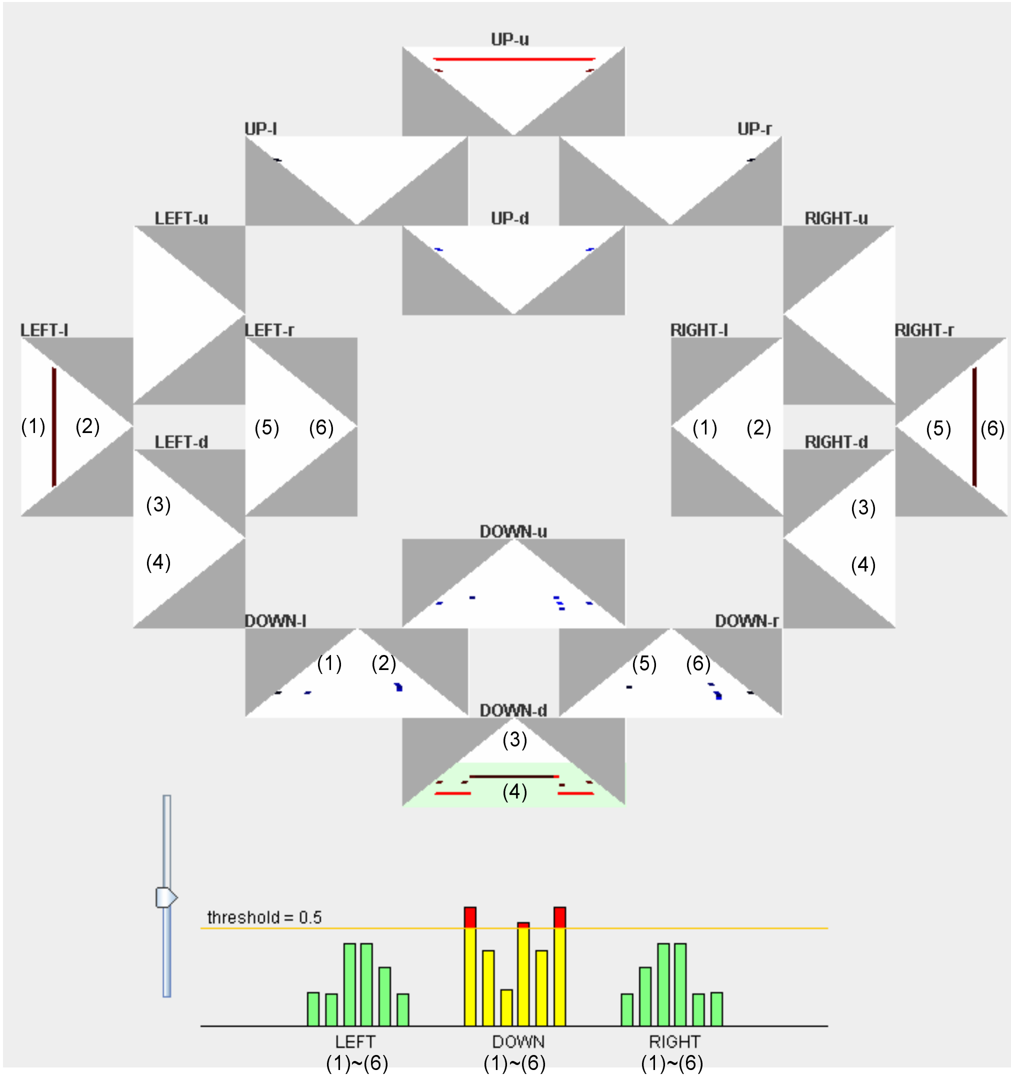

Figure 9. System operation results using artificial simple images.

sion system for far distance obstacle detection. In *IEEE Intelligent Vehicles Symposium*, pages 292–297, 2004.

[6] N. Shima, M. Babasaki, Y. Akidzuki, K. F. Honda, T. Higuchi, H. Higashida, and R. Nakamura. Fusion sensor for driving assistance system. (17):35–44, 2001.

[7] M. Yokohama and T. Poggio. A contour-based moving object detection and tracking. In *Proceeding of Second Joint IEEE International Workshop on Visual Surveilance and Performance Evalaution of Tracking and Surveilance (VS-PETS)*, pages 271–276, Beijing, China, Oct. 2005.

[8] M. Yoshida and H. Hayashi. Organization of cell assemblies that code temporal sequences in a hippocampal CA3-CA1 model. In *Proc. Int. Joint Conf. on Neural Networks (IJCNN)*, pages 495–500, 2004.