Detecting Approaching Cars via Artificial Insect Vision

Naoya Ohta
Department of Computer Science, Gunma University
1-5-1 Tenjin-cho, Kiryu, Gunma 376-8515 Japan
ohta@cs.gunma-u.ac.jp

Abstract

The application of computer vision or image processing technology is typically limited to expensive machines in big facilities. One reason for this phenomenon is the cost of hardware required by this technology. In order to develop computer vision algorithms runnable on inexpensive hardware, we look at the vision system of insects, and discuss four ideas we could take into account when developing such algorithms. On the basis of this discussion, we specify a design for an approaching-car detection system for car driver assistance. This system is attached to the rear of a car, and faces backwards. If other cars approach from the rear, the system informs the driver. We investigated the performance of the system using various experiments and demonstrated its effectiveness. The amount of computation the system must carry out is approximately one ten-thousandth of that required by a conventional image processing algorithm. It is executable on a small and inexpensive processor for embedding purposes.

1 Introduction

Technology of computer vision or image processing is increasingly being applied to real world problems and is utilized in many practical tasks. For instance, thanks to that technology, machines can detect defects of products in factories, read post codes on letters, and replace advertisement signs with others in tennis game TV programs [5]. However, we may also observe that application of this technology is largely limited to expensive machines in big facilities. We almost never see consumer machines, such as television sets or VCRs, being equipped with computer vision technology. The biggest reason for this phenomenon lies in the cost of the required hardware.

As images are large repositories of data, it is natural that fast, expensive computers are required. For the applications mentioned above, it is not a crucial problem because the cost is not the first priority in such applications. However, for consumer products, this problem is crucial because the cost is very important. Therefore, if we wish to develop consumer products equipped with computer vision technology, we have to develop very inexpensive computer vision algorithms executable by small and slow computers.

Historically speaking, computer vision research has been linked with the analysis of human vision [4]. This might be the reason why most computer vision research seems to be aimed at the realization of the same kind of functions as those of humans. However, humans are not the only life which has vision. Insects have vision. This kind of vision clearly provides crucial information for their lives. At the same time, these vision systems must be implemented on very simple hardware or neural systems. Therefore, if we look at those vision systems, we might be able to get some hints as to how we might develop simple computer vision algorithms runnable on small and inexpensive hardware. Of course, even if we find such algorithms, they would be useless if they do not provide valuable functions. However, it is known that houseflies stabilize their flight and control landing by vision [1]. There must be many consumer products to which these functions are applicable.

In this paper, we discuss an approaching-car detection system inspired by insect vision. This system is designed to be set on a car facing backwards. When other cars approach from the rear, it visually detects them and warns the driver. The point of this system is that it requires only very inexpensive hardware.

2 Artificial Insect Vision

There have been many studies inspired by insect vision. One of the most successful examples is the study reported in [2]. In that study, a mobile robot was actually built, which was able to move around in a room avoiding obstacles using an insect-like vision system. The focus of this study lies in an attempt to understand the mechanisms of insect vision, copying them faithfully onto artificial machines. As our purpose is to make inexpensive vision systems, we consider insect vision from a somewhat different point of view.

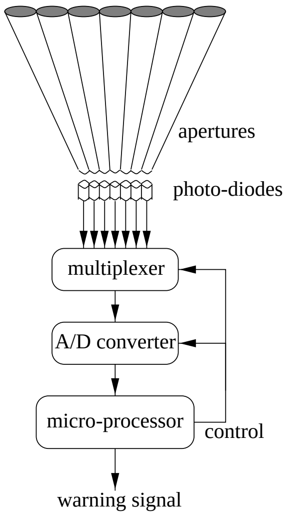


Figure 1. Hardware of the system.

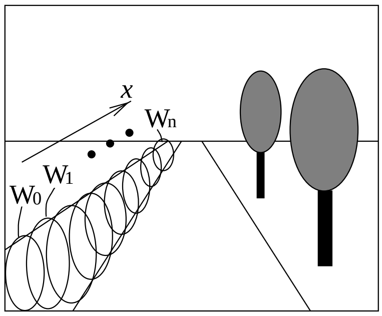


Figure 2. Receptive fields of apertures.

We do not attempt to copy the data processing procedures of insect vision. Rather, we imitate mechanisms of the insect vision if they reduce the amount of computation. Specifically, we reconsider our vision systems from the following points of view.

(a) Utilization of low resolution images:

The fact that insects use compound eyes suggests that high resolution images are sometimes not essential to get valuable information from vision. We consider usage of low resolution images because they reduce drastically the amount of data to be processed.

(b) Utilization of imaging devices other than video cameras:

Video cameras were originally designed to get images akin to those humans see. However, other imaging devices like photo diodes whose fields of view are controlled by apertures could provide images suitable for low price systems.

(c) Utilization of optical image processing:

Optical systems can also process images. It is well known that frequency filtering of images can be easily carried out with a lens system [7]. So, we should consider whether we can use optical image processing for our systems.

(d) Utilization of analog circuits for computation:

Digital circuits are usually used for data processing in computer vision systems. However, analog circuits could be advantageous when computing the difference, sum or time delay of signals from imaging devices.

By considering our computer vision systems from the above points of view, we might be able to design inexpensive systems. Of course, we need not adopt all of those ideas. We adopt some of them only when they reduce the cost of our systems. We used the first two ideas above for our approaching-car detection system. We call this designing strategy *Artificial Insect Vision* in this report.

Figure 1 shows the hardware of the approaching-car detection system we designed following the Artificial Insect Vision strategy. A special imaging device, whose fields of view are controlled by cylindrical apertures, provides intensity signals for car detection. The number of apertures or signals is several tens. So, this device reduces the amount of data to be processed. The intensity signals are converted in digital form by an A/D converter and are processed by a microprocessor. This microprocessor can be a small and inexpensive one usable in embedded systems. We actually used digital circuits for this part, but it could have been realized by analog circuits. We leave this subject for the future work.

3 Car Detection System

The back side direction can be a dangerous zone because it is hard for the driver to see. If the driver changes lane without noticing other cars coming, it can obviously cause a collision. Therefore, a system that could monitor that zone and warn of other cars coming would be useful. Many vision systems monitoring the surroundings of a car have been reported, and they often require high speed processors specially designed for image processing [3]. As stated in the previous section, our system requires only a small processor, owing to an insect-like imaging device.

The principle of car detection is quite simple. When our car moves forward, objects in the back side views move (relatively) backwards. However, if there are cars approaching our car, forward movement is observed. The system detects this movement using optical flow and therefore gauges the existence of other cars.

The apertures of the imaging device in Fig. 1 are set so that their field of view cover the adjacent lane. The ellipses in Fig. 2, indicated by W_x ($x = 0 \ldots n$), show the field of view of the apertures. Let $S_{[t,x]}$ denote the intensity signal obtained from the ellipse W_x at the time t. Approaching cars are detected from $S_{[t,x]}$ by the following algorithm.

— Step 1 —

We compute the space derivative $S_{X[t,x]}$ and the time derivative $S_{T[t,x]}$ of the intensity $S_{[t,x]}$ by the following equations.

$$S_{X[t,x]} = \frac{S_{[t,x+1]} - S_{[t,x]} + S_{[t+1,x+1]} - S_{[t+1,x]}}{2} \quad (1)$$

$$S_{T[t,x]} = \frac{S_{[t+1,x]} - S_{[t,x]} + S_{[t+1,x+1]} - S_{[t,x+1]}}{2} \quad (2)$$

— Step 2 —

We prepare a data array $Q_{[t,x]}$, which stores states. If the condition

$$|S_{X[t,x]}| < \theta_1 \quad (3)$$

holds, we set $Q_{[t,x]}$ to state A. In the above equation, θ_1 is a threshold, which is set to 2.0 in the experiments. This condition indicates that the space derivative is too small to compute reliable optical flow [6]. It also suggests that no cars exist because car bodies usually make an intensity texture and a large space derivative is observed in such the areas. For the part where the condition (3) does not hold, we apply the following procedures.

— Step 3 —

We compute 1-dimensional optical flow $v_{[t,x]}$ by

$$v_{[t,x]} = -\frac{S_{T[t,x]}}{S_{X[t,x]}}, \quad (4)$$

and check the condition

$$v_{[t,x]} \geq 0. \quad (5)$$

If the condition (5) holds, we set $Q_{[t,x]}$ to state B, otherwise to state C. State C indicates that "expanding" optical flow was detected, which can be interpreted as approaching cars exist. On the other hand, state B shows that "contracting" optical flow was detected, which must be generated by patterns on the road surface such as a zebra crossing, or slower cars than ours. We thus summarize the meaning of each state as follows.

State A: No cars.

State B: No cars, or cars going apart.

State C: Approaching cars.

— Step 4 —

We look at the part of the array $Q_{[t,x]}$ sectioned by the current time t_0 and a passed time t_1, and count the number of states of type C in that part. If this number is greater than a threshold θ_2, the system decides the existence of approaching cars and outputs the warning signal. In the experiments, we set θ_2 to 60 and the time span between t_0 and t_1 to 10 frames. The function of this procedure is to integrate the states in the array $Q_{[t,x]}$ and to provide a stable output. Individual states of $Q_{[t,x]}$ are unstable because they are computed using only local information.

Figure 3. Gaussian weightings on experimental images.

4 Experiments

We conducted experiments in order to investigate the effectiveness of our system design. In order to get the intensity signals $S_{[t,x]}$, it is best to actually make the imaging device in Fig. 1, but as the first step, we simulated the function of the imaging device by accumulating pixel values of regular video images with Gaussian weightings, because of the ease of conducting such experiments. The receptive fields of the imaging device, or Gaussian weightings in this case, are shown in Fig. 3. The ellipses show the standard deviation regions of the Gaussian weightings. The ratio of the short axis to the long axis of all ellipses is 0.2. The number of ellipses is 50. We decided this shape and the number of the ellipses on the basis of a preliminary experiment.

Figure 4 shows the experimental result for a daytime scene. The original image sequence, which was taken by a regular video camera, consists of 300 frames, from which 4 frames are shown in Fig. 4 (a). The intensity $S_{[t,x]}$, the state $Q_{[t,x]}$ and the final decision are shown in Fig. 4 (b), (c) and (d), respectively. In these images, the horizontal axis is time or frame t. White lines marked in the bottom indicate 10 frame intervals. For Fig. 4 (b) and (c), the vertical axis is the position x of the receptive field W_x. In Fig. 4 (a) and (c), state A, B, and C are expressed by gray, white and black, respectively. In Fig. 4 (d), the white regions indicate that the system detected approaching cars. In this daytime scene, a car approaches in the first part of the sequence, then goes back once, and gets closer again to pass our car. The

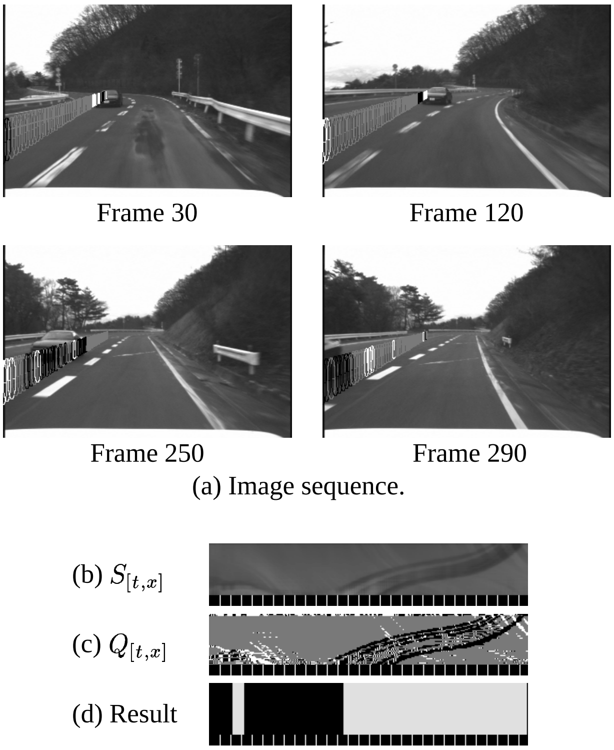


Figure 4. Car detection result 1 (daytime)

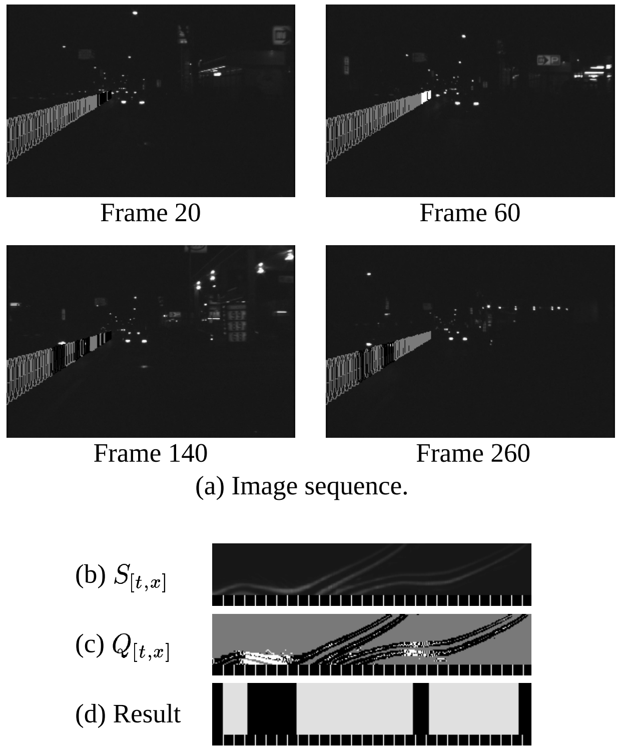


Figure 5. Car detection result 2 (night).

system detected this situation correctly.

A result of the processing of night-time images is shown in Fig. 5. The images in Fig. 5 (a), (b), (c) and (d) show the same information as in Fig. 4. In this scene, two cars pass our car. The headlights of the cars yield good contrast in the intensity signals $S_{[t,x]}$, and the system stably detects the approaching cars. There are two gaps in the middle part of the detection result of Fig. 5 (d). In these gaps, the cars actually go back.

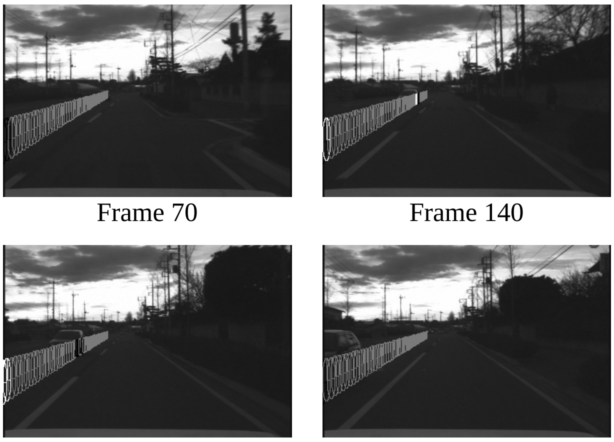
We show some scenes for which the system failed to correctly give a warning. One is a twilight scene shown in Fig. 6. In this scene, the road has a very low constant in the intensity because the sky is so bright and the camera controls its gain to meet the sky's brightness. As a result, the system fails to detect the passing car. If we set the threshold θ_1 in Eq. (3) to 0.5 instead of 2.0 in order to increase the sensitivity, the system outputs the correct result as shown in Fig. 6 (e) and (f). This suggests that we can cope with this problem by controlling the gain according to the intensity signals $S_{[t,x]}$.

The other example where the system failed is a scene of curved roads, shown in Fig. 7. The principle for car detection stated in the previous section is valid only when the road is straight and our car goes straight. Therefore, it is easily surmised that the system will be confused when our car travels on a curved road. In the scene of Fig. 7, the system output false alarms from the middle of the sequence because our car started to turn right at that point. The rotation of our car generated approaching movement of the white lines, and the system detected this movement as approaching cars. However, we can cope with this problem by changing the positions of the receptive fields W_x according to the steering information. We are now working on this problem, and can show only a preliminary result in Fig 8. Figure 8 (a) shows the arrangement of the receptive fields and (b) is the state $Q_{[t,x]}$ for the last 150 frames of the same image sequence in Fig. 7. We observe that the false movement of the white lines has disappeared.

The biggest advantage of the car detection system proposed in this paper is the small computing cost. In order to show this advantage, we measured the processing times, which are shown in Table 1. The figures in the table are the times in second for processing one frame of the image sequence by an Intel Pentium II 400MHz processor. The first row, which is labeled "Imaging simulation", is the time needed to simulate the function of the imaging device,

Table 1. Computation time par frame.

Process	Time in second
Imaging simulation	1.22
Car detection	3.46×10^{-5}
Optical flow computation	1.44



Frame 70 Frame 140

Frame 200 Frame 270

(a) Image sequence.

(b) $S_{[t,x]}$

(c) $Q_{[t,x]}$

(d) Result

(e) $Q_{[t,x]}$

(f) Result

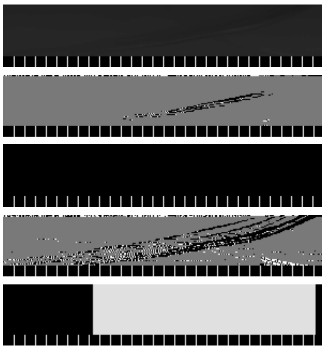
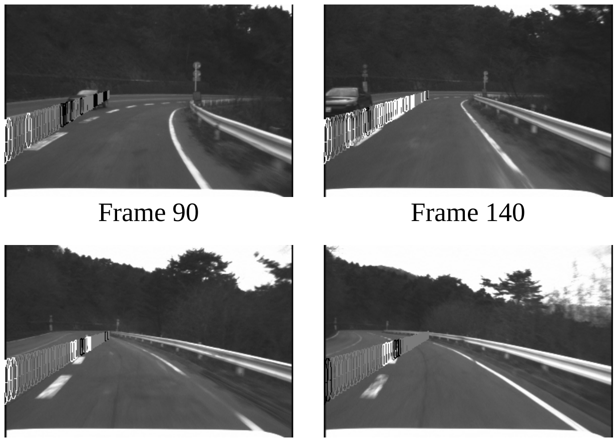


Figure 6. Car detection result 3 (twilight).

namely, to accumulate pixel values of input images with the Gaussian weightings in order to obtain the intensity signals $S_{[t,x]}$. The second row, labeled "Car detection", is for the car detection process described in Section 3, which the processor in Fig. 1 is supposed to actually compute. The third row, labeled "Optical flow computation", is the time for computing optical flow at 56×40 (2240) points on the image by a usual optical flow computation method [6] (Fig. 9). We showed this figure only comparison purposes as an example of processing time of conventional image processing algorithms. Comparing the second row with the third one, we observe that the computing cost of the proposed method is smaller in the order of 10^{-5} than that of the conventional optical flow computation. This indicates that a small and inexpensive processor can easily carry out our method.

5 Summary and Conclusions

In this paper, we first discussed what kind of hints we can get from the vision system of insects for the purpose of developing computer vision algorithms executable on small and inexpensive hardware. We summarized the hints as four



Frame 90 Frame 140

Frame 230 Frame 290

(a) Image sequence.

(b) $S_{[t,x]}$

(c) $Q_{[t,x]}$

(d) Result

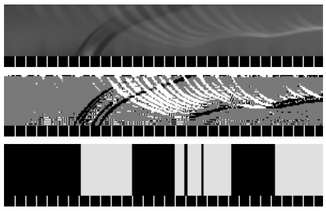


Figure 7. Car detection result 4 (curved road)

points we should note when designing such a vision system. Then, following such a design strategy, which we termed Artificial Insect Vision, we designed an approaching-car detection system. The experiments show that the system provides a good performance at a small computing cost. The amount of computation the algorithm requires is in the order of 10^{-5} of that required by a conventional image processing algorithm.

There are two main issues left for the future research in order to make the system more practical. One is the problem for curved roads. As mentioned in the section for the experiments, we are trying to solve this problem by changing positions of the receptive fields of the imaging device according to the information from the steering wheel of the car. If we assume the road surface is flat, patterns on the road move along an ellipse or conic on the image, when the car is rotating. Therefore, if we align the receptive fields along that ellipse, the system can distinguish optical flow generated by approaching cars from that due to road patterns.

The second issue is to actually construct the system—especially the imaging device. We have found it desirable that the receptive field of a photo detector of the imaging device has a Gaussian-like sensitivity, as we assumed in the experiments. The problem is whether we can realize these characteristics using a cylindrical aperture. Preliminary ex-

(a) Receptive fields along a curved line.

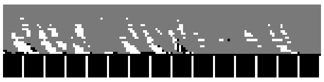
(b) $Q_{[t,x]}$ 

Figure 8. Experiment with curved receptive fields.

Figure 9. Optical flow by a conventional algorithm.

periments suggest the answer is in the affirmative.

Another issue for the future work, aside from approaching car detection, is to investigate what other kind of systems we can design on the basis of the Artificial Insect Vision strategy. We are planning, for instance, a collision avoidance system and a rotation sensor, but we cannot state much about these yet, as the research has just commenced.

Acknowledgments

The author thanks Professor Kenichi Kanatani of Okayama University and Professor Michael J. Brooks of University of Adelaide, who discussed technical issues with him and helped to correct the English in this paper. The author also thanks Mr. Kazutaka Niijima of Toshiba Corporation and Mr. Hirotaka Yamamoto of Gunma University for conducting the experiments.

References

[1] W. Bialek, M. Potters, D. Ruderman and R. R. de Ruyter van Steveninck, “Visual Computation: A Fly’s Eye View,” *Proceedings of the 4th NEC Research Symposium*, May 1993, pp.7–22.

[2] N. Franceschini, J. M. Pichon and C. Blanes, “From Insect Vision to Robot Vision,” *Philosophical Transactions of the Royal Society of London*, Ser. B, No. 337, 1992, pp.283–294.

[3] S. Kyo, T. Koga, K. Sakurai and S. Okazaki, “A robust vehicle detecting and tracking system for wet weather conditions using the IMAP-VISION Image Processing Board,” *Proceedings of IEEE/IEEJ/JSAI International Conference on Intelligent Transportation Systems*, Tokyo Japan, Oct. 1999, pp. 423–428.

[4] D. Marr, *Vision : A computational investigation into the human representation and processing of visual information*, W. H. Freeman and Co., 1982.

[5] G. Medioni, “Using computer vision in real applications: Two success stories,” *Proceeding of IAPR Workshop on Machine Vision Applications*, Tokyo, Nov. 1996, pp. 352–360.

[6] N. Ohta, “Image movement detection with reliability indices,” *IEICE Transactions*, Vol. E74, No. 10, Oct. 1991, pp. 3379–3388.

[7] R. G. Wilson, *Fourier series and optical transform techniques in contemporary optics : An introduction*, John Wiley and Sons, 1995.