15-5

MVA2007 IAPR Conference on Machine Vision Applications, May 16-18, 2007, Tokyo, JAPAN

Overtaking Vehicle Detection Method and Its Implementation Using IMAPCAR Highly Parallel Image Processor

Kazuyuki Sakurai, Shorin Kyo and Shin'ichiro Okazaki
Media and Information Research Labs,
NEC Corporation
1753, Shimonumabe, Nakahara-Ku, Kawasaki, Kanagawa 211-8666, Japan
k-sakurai@bq.jp.nec.com, s-kyo@cq.jp.nec.com, s-okazaki@cq.jp.nec.com

Abstract

This paper describes a vision-based overtaking vehicle detection method for driver assistance systems and its implementation using IMAPCAR, a highly parallel SIMD linear array processor. The proposed overtaking vehicle detection method is based on optical flows detected by block matching using SAD. The optical flow detection is made robust through its vanishing point detection. As a result, stable performance of the overtaking vehicle detection is confirmed and video-rate (33frames/s) implementation is accomplished by IMAPCAR.

1 Introduction

Recently, various driver assistance systems have been put to practical use. These systems are equipped with in-vehicle sensors for detecting objects around the vehicle. Among the various types of in-vehicle sensors, the camera is one of the most important sensors because of its ability to capture a large amount of information [1]. Many functions for driver assistance systems can be realized by image recognition, for example, lane detection, vehicle detection and traffic sign detection. Overtaking vehicle detection application is one of the useful applications, which can be used for approaching vehicle warning when driver is changing lane.

A previously proposed method to this application is based on optical flow [2], where vanishing point of flow is first detected as the cross section point of detected lane mark lines, and then optical flow is detected along each straight line expanded from that vanishing point. Comparing with [2], in this paper, we improved the quality of optical flow detection, first by using a more robust vanishing point detection method, and second by rejecting ambiguous flow detection results based on their reliability scores.

Such image recognition tasks generally require high processing power as much as high power General Purpose Processors (GPPs) have. However, in-vehicle image processors are required to emit little heat to facilitate in-vehicle placement and GPP consumes too much power. Another approach is to use ASIC (Application Specific Integrated Circuit), while ASIC has little flexibility so that it must be designed exclusively for every function and costs very much. Therefore it is necessary to use ASSP (Application Specific Standard Product) for in-vehicle image recognition. IMAPCAR [3], a highly parallel SIMD linear array processor is designed as ASSP for it. It is another issue how to take advantage of its parallel SIMD architecture in implementation of the proposed overtaking vehicle detection method using IMAPCAR.

Comparing with previously proposed in-vehicle multi-processor architecture such as Visconti, which is based on three processor cores each equipped with a separate control unit [4], IMAPCAR contains less control circuit while more processing elements. As in many cases one single control flow will be sufficient to implement time consuming image processing tasks such as optical flow detection, fully exploiting the existing high degree of data level parallelism is more important, thus a processor such as IMAPCAR will be a better choice.

This paper describes and evaluates an overtaking vehicle detection method and its implementation using IMAPCAR. As a result of our evaluation, we found that the overtaking vehicle detection method is robust, and its efficient implementation can be done by using IMAPCAR, with a power efficiency that is 66 to 132 times superior to that when using a 3 GHz GPP.

2 Overtaking Vehicle Detection Method

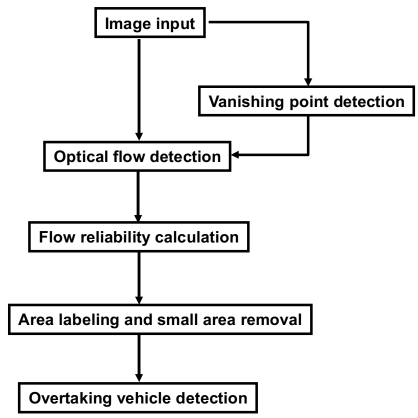
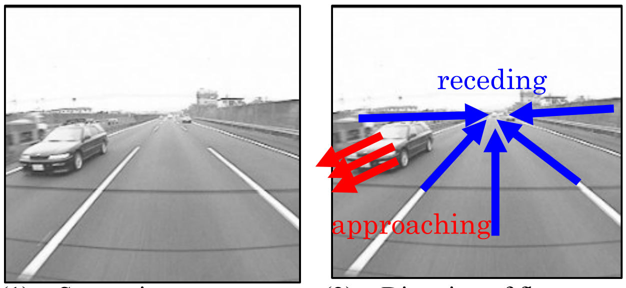


Figure 1. Overtaking vehicle detection method.

Figure 1 shows the proposed overtaking vehicle detection method, and Figure 2 (1) shows a source road image example. The overtaking vehicle detection method is based on optical flow and uses the fact that the optical flow of overtaking vehicles is in an approaching direction, while that of the background and other vehicles is in a receding direction (Figure 2 (2)).

2.1 Optical flow detection

Optical flow is detected by block matching using SAD (Sum of Absolute Difference). An alternative approach for calculating optical flow is to use spatio-temporal filters. However, while the later method has a lower processing cost, its underlying assumption of infinitesimal differences



(1) Source image (2) Direction of flow

Figure 2. Source image and direction of flow.

between two consecutive temporal images is not valid for video-rate images of in-vehicle cameras used to capture high-speed moving objects.

One of the difficulties when detecting optical flow is that flow direction in homogeneous area or single orientation texture areas is ambiguous. In road images in particular, there is a large area of homogeneous road surface, and some areas that happen to have the same direction seem to be overtaking vehicles even when they are not. Such areas are often mistakenly detected as overtaking vehicle areas.

To avoid this kind of problem, we introduced a method to improve the reliability of flow detection. Flow reliability is improved by identifying mutual corresponding pixel blocks between two temporal images using not only the minimum SAD value but also the information of the second minimum SAD and the average SAD within each search area, each normalized by the average intensity of pixels within each block. Figure 3 shows flow reliability in which the brighter pixels have a higher reliability and the darker ones a lower reliability. The flows that have a higher reliability than a given threshold are kept and all others are discarded as noise. The threshold is determined by experiments. Figure 4 shows detected flows. In this figure the yellow areas shows optical flows in the approaching direction, and the blue areas optical flows in the receding direction.

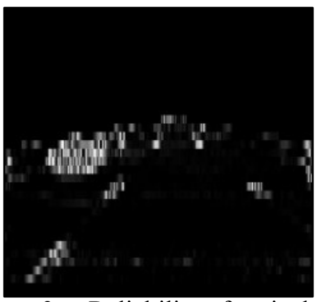


Figure 3. Reliability of optical flow.

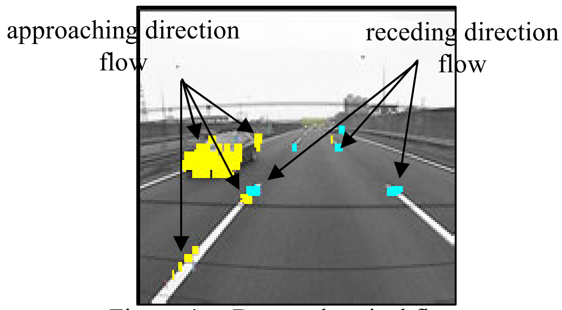


Figure 4. Detected optical flows.

Further, the areas of connected pixel locations having approaching direction flows are detected by connect component labeling, while small areas are discarded. As a result, each area with a unique label is detected as an overtaking vehicle area. Moreover, the robustness of overtaking vehicle detection is improved by tracking detected vehicle areas through time.

2.2 Vanishing point detection

The optical flow to be detected, especially for highway scenes, has in general the same direction as straight lines expanding from the so-called vanishing point, which corresponds to the vehicle's own direction, and coincides almost exactly with the vanishing point of lane markers in most cases. We therefore decided to also detect nearby lane markers and their vanishing point, and sought to further increase the robustness of optical flow detection by limiting optical flow directions along expanded lines from the vanishing point (Figure 5).

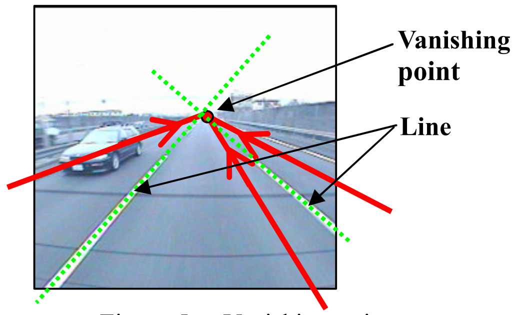


Figure 5. Vanishing point.

Our vanishing point detection method is composed of three steps, 1) edge point detection, 2) Hough transform, and 3) reverse Hough transform. During edge point detection, points with a relatively large horizontal gradient are detected. Hough transform detects straight lines corresponding to nearby lane markers and other objects that are in same direction with lane markers, such as guard rails and road area side lines. Reverse Hough transform draws the detected lines in the road image space with their Hough vote, and detects the line crossing point with the largest vote collection as vanishing point.

In the ideal case, all the detected lines cross at a single point (vanishing point). However this is not the case in practice, so that the detected vanishing point position is incorrect (Figure 6). Therefore, we introduced the following two processes for the line drawn image space.

1. Large kernel smoothing of vote collection
2. Evaluation of the variety of line directions

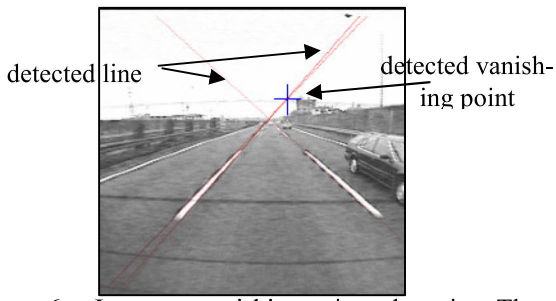


Figure 6. Incorrect vanishing points detection. The red lines are the detected lines, and the blue cross sign is the detected vanishing point.

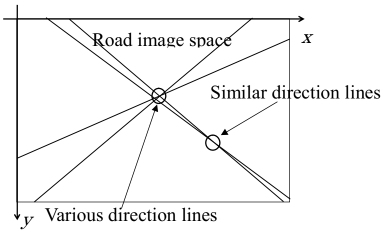


Figure 7. Line directions at crossing points.

Large kernel smoothing of vote collection results in larger vote collection at the vanishing point through summation of vote collections in nearby areas. Evaluation of the variety of line directions is based on the assumption that various direction lines cross at the true vanishing point, while similar direction lines cross at fake vanishing points (Figure.7). To make vote collection larger at points that have lines in various directions, especially left and right lane makers, the vote collections at all points are multiplied by evaluation coefficient E as follows.

$$E = \frac{2 \cdot \min(S_l, S_r)}{S_l + S_r}$$

S_l is the amount of vote collection in the lower left area and S_r is that in the lower right area, as shown in Figure 8. E is larger at points that have lines both in the lower left and in the lower right area, and it is smaller at points that have lines only in the lower side area. Therefore, E evaluates the variety of line directions. Figure 9 shows that vote collection in the road image space is improved by the proposed evaluation coefficient.

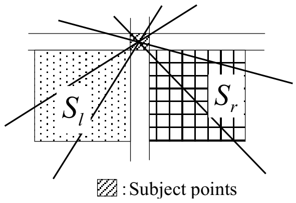


Figure 8. Evaluation of variety of line directions.

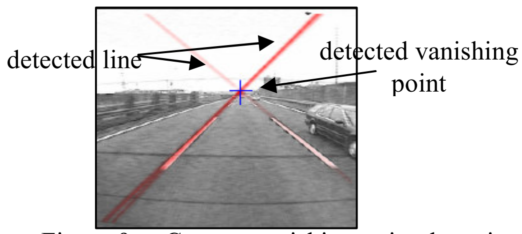


Figure 9. Correct vanishing point detection.

3 Implementation of Overtaking Vehicle Detection using IMAPCAR

3.1 IMAPCAR Image Processor

Figure 10 shows a simplified block diagram of the IMAPCAR highly parallel SIMD linear array processor and a photo of its PCI test board. IMAPCAR is fabricated using a 0.13 µm CMOS process and integrates 26.8 million transistors, including 128 8-bit 4-way VLIW RISC PEs (Processing Elements) each equipped with a 24-bit MAC (Multiply Add Accumulate), 256 KB (2 KB/PE) of data RAM (IMEM), one CP (16-bit RISC Control Processor) with a 32 KB instruction cache and 2 KB data cache, and a DMA engine for data transfer between the IMEM and external SSRAM. A shift-register configuration is used for transferring video data in parallel with PE operation into the IMEM or external SSRAM. The LSI is packaged in a 500-pin TBGA and satisfies the temperature range requirement (-40 to +85 degrees Celsius) for automotive use. The application-level power consumption is estimated to average approximately 1 to 2 watts during operation at 100 MHz.

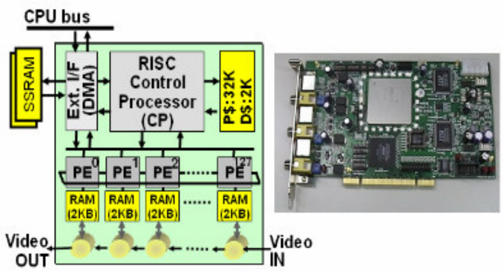


Figure 10. IMAPCAR.

Figure 11 shows the IMAPCAR programming model. Column-wise mapping of one image to each PE is assumed, and the collection of all PE local memories (IMEM) is called the 2-D memory plane, where source, destination, and work images can be explicitly allocated by using the 1DC (One Dimensional C) programming language, which is a straightforward parallel extension of C. Each PE can perform access to different pixel locations in each row (indirect addressing capability), which, as described later, greatly facilitates parallelization of some image tasks containing irregular memory accesses.

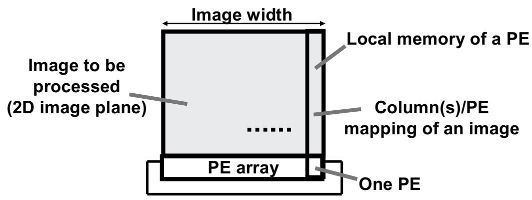


Figure 11. Image stored in IMAPCAR.

3.2 Implementation using IMAPCAR

The following are two important modules for the implementation of the overtaking vehicle detection algorithm described in section 2: 1) Optical flow detection and flow reliability examination, and 2) Hough transform for lane marker detection. Reverse Hough transform can be implemented in the same way as Hough transform. This subsection presents ways to implement these modules taking advantage of IMAPCAR characteristics.

As a characteristic of IMAPCAR is the parallel operation of PEs as described in subsection 3.1, we implemented optical flow detection using IMAPCAR by first generating a shifted image by moving all pixels in the previous temporal image to an offset of (-dx, -dy). Note that (dx, dy) is within the block-matching search area. Next, the differences between the shifted image and the current temporal image are calculated pixel-wise (Figure 12). The value of each pixel in the difference image is now the AD (Absolute Difference). The SAD can be calculated by accumulating pixel values in each local area of the difference image. This process is done for each (dx, dy) value. Then, the normalized minimum SAD, second minimum SAD and average SAD are calculated at each pixel location of the current image, and based on these three values, the optical flow of each pixel location and its reliability are calculated. As the above processes can all be performed independently for each pixel, parallel processing can be easily achieved by allocating processing at each pixel of one row to each PE. Step-by-step image shifting also minimizes the amount of data transfer between PEs. Moreover, the storage space requirements for the three different types of SAD values are solved by swapping data between the IMEM and external SSRAM using the DMA engine described in subsection 3.1, through which data transfer is done simultaneously with PE operation.

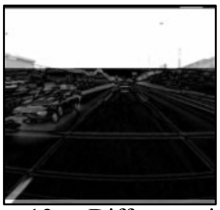


Figure 12. Difference image.

For detection of lane markers using the Hough transform, first, feature points are detected using a local edge filter and are selected using a threshold value. Then, these feature points are collected to the CP. When a line is expressed by the equation y = a * x + b, the Hough space is a parameter space (a, b) and the locus about feature point (x1, y1) is a line expressed by b = x1 * a + y1 (Fig. 13). Implementation of the Hough transform using IMAPCAR is done by allocating the Hough space in the PEs' local memory (Figure 13) so that the "a" axis is along the PE array. Feature points collected by the CP are then broadcast to all PEs, and each PE calculates in parallel the "b" value and transfers its vote to the corresponding index in its local memory using the indirect addressing capability described in subsection 3.1. In this way, the voting process, which accounts for most of the time required for the Hough transform, can be efficiently implemented in parallel by all PEs.

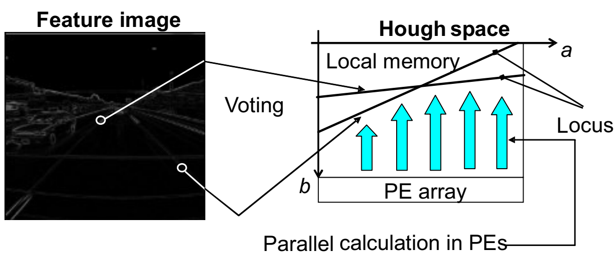


Figure 13. Hough transform.

4 Evaluation

Figure 14 shows an example of the overtaking vehicle detection results. Table 1 shows evaluation result on about 30 minutes video in Tomei highway under wet weather condition, which contains recall rate and false positive detection times with or without vanishing point detection. Table 1 shows that the performance of the detection is made robust through the vanishing point detection.

Figure 14. Detection result.

Table 1. Performance of detection.

	recall	false positive
With vanishing point detection	98 %	0 times
Without vanishing point detection	91 %	1 times

The processing times of compiler generated code for both the IMAPCAR and GPP are listed in Table 2. The image size is 256x240, the block size of SAD is 7x7 and the search area of minimum SAD is 11x11. For the GPP, only the optical flow detection and vanishing point detection processes, which take the most time, are evaluated for comparison. Table 1 shows that IMAPCAR is 2.73 times faster than a 3 GHz GPP. IMAPCAR is also 68 to 136 times more power efficient since the power consumption of IMAPCAR is approximately 2 W while that of the Pentium 4 is approximately 50 to 100 W.

Table 2. Processing time.

	IMAPCAR @100 MHz	GPP @3 GHz
Optical flow detection	21.89 ms	55.03 ms
Vanishing point detection	5.25 ms	19.10 ms
Vehicle area detection	4.43 ms	------
Total	31.57 ms	>74.1ms

5 Conclusion

This paper presents an overtaking vehicle detection method and its implementation on IMAPCAR. The results show that the proposed method is robust and IMAPCAR has advantages for the implementation in terms of performance and low power consumption.

References

[1] M. Bertozzi, et. al.: "Artificial Vision in Road Vehicle", *Proc. of IEEE*, vol.90, no.7, pp.1258-1271, 2002.

[2] Y. Taniguchi and N. Ishikawa: "Automatic Rear and Side Surveillance System Using Image Processing", *Proc. of SSII2000*, pp.23-28, 2000 (in Japanese).

[3] S. Okazaki, S. Kyo and F. Hidano: "IMAPCAR: A Highly parallel integrated memory array processor for in-vehicle image recognition applications", *Proc. of ITS World Congress*, ID-1744, 2006.

[4] T.Miyamori, et. al.: "Development of Image Recognition Processor Based on Configurable Processor", *Journal of Robotics and Mechatronics*, vol.17, no.4, pp.437-446, 2005.