Realtime Multiple Object Tracking Based on Optical Flows

Shinya Yamamoto, Yasushi Mae, Yoshiaki Shirai, and Jun Miura

Department of Mechanical Engineering for Computer-Controlled Machinery
Osaka University, 2-1, Yamadaoka, Suita, Osaka 565, JAPAN

Abstract

This paper describes real time object tracking by extracting optical flows from a sequence of images. Optical flows are calculated based on the generalized gradient model. Our method detects a moving object, and tracks it from the opticalflow data. Assuming that pixels corresponding to the same object have similar flow vectors, we extract a connected region with similar flow vectors. We apply this method to multiple object tracking. When two objects overlap, foreground object is recognized, and each object is tracked without confusion.

Optical flow extraction and object tracking are executed in realtime by a special image processor.

1 Introduction

Visual object tracking is useful for many applications, such as surveillance and autonomous vehicle navigation. This paper describes extraction of optical flows from a sequence of images and tracking of a moving object from the flow vectors in realtime.

Several techniques for calculating optical flow are proposed[1]. Most methods need much computational costs. For instance, Horn and Schunck's method [2] requires iterations for regularization.

For a limited number of specified points in a image the optical flow can be extracted by correlation. Tracking of road boundaries[6], for example, is actually extracting the motion of the boundaries by correlation using multiple transputers. Special hardware is developed for computing optical flows by correlation and for segmenting the flows [7]. In Oxford University, a reactive camera head is developed [8] which extract a moving object and track it while controlling pan and tilt. One problem with the correlation method is the computation cost, which make it difficult to extract dense flows.

A powerful chip has been employed to compute approximate correlations of many points (100) and determine the extreme[5]. However, the number is not enough to obtain dense flows. Moreover, the extracted flow is not reliable if the correlation function has not a clear peak.

We calculate the optical flow based on the generalized gradient model [3],[4], which measures spatial and temporal intensity gradients. The computational costs of this method is relatively low because it does not need iteration. Assuming that pixels which correspond to the same object have similar flow vectors, we extract a region with similar flow vectors as an object region and track it by updating such a region in each image. We apply this method to multiple object tracking.

We detect objects which come into the image in order and track them separately. If they overlap in the image, each object is tracked without confusion by considering optical flow data.

We use a special image processor[9],[10] to realize optical flow extraction and object tracking in realtime.

Section2 in this paper describes the theory of extracting optical flow Section3 describes the method of object tracking, and Section4 describes the configuration of the image processor.

2 Extracting optical flow

The optical flow extraction is based on the generalized gradient model.

In a typical gradient method the relation between a sequence of two-dimensional image, $f(x, y, t)$, and the velocity u and v at (x, y, t) is represented by the following equation

$$u \cdot f_x + v \cdot f_y + f_t = 0 \tag{1}$$

where subscripts denote partial differentiation.

Orientation-selective spatial Gaussian filters are used to obtain the constraint equations. We apply two orientation-selective spatial filters to the original

IEEE International Conference
on Robotics and Automation
0-7803-1965-6/95 $4.00 ©1995 IEEE

image: one is sensitive to vertical edges, the other to horizontal edges.

Let $g(x,y)$ and $h(x,y)$ denote the two filters and $g \otimes f$ denote the convolution of $f(x,y,t)$ and g. The following constraint equations are given.

$$u \cdot g \otimes f_x + v \cdot g \otimes f_y + g \otimes f_t = 0 \quad (2)$$
$$u \cdot h \otimes f_x + v \cdot h \otimes f_y + h \otimes f_t = 0 \quad (3)$$

In order to decrease the effect of noise, a temporal filter $p(t)$ is applied for smoothing. The equations above are modified by convolving the temporal filter.

$$u \cdot g_x \otimes f \otimes p + v \cdot g_y \otimes f \otimes p + g \otimes f \otimes p_t = 0 \quad (4)$$
$$u \cdot h_x \otimes f \otimes p + v \cdot h_y \otimes f \otimes p + h \otimes f \otimes p_t = 0 \quad (5)$$

where p_t is the derivative of temporal filter p. From the two filtered images we derive two indepentdent constraint equations at each pixel and by solving the equations we obtain the optical flow vector.

The reliability of the velocity measurement at each pixel location is low unless the following two conditions are satisfied.

$$(g_x \otimes f \otimes p)^2 + (g_y \otimes f \otimes p)^2 + (h_x \otimes f \otimes p)^2 + (h_y \otimes f \otimes p)^2 > n_1 \quad (6)$$

where n_1 is a predetermined gradient threshold value. This means that the spatial gradient of the image must be large.

$$|(g_x \otimes f \otimes p)(h_y \otimes f \otimes p) - (g_y \otimes f \otimes p)(h_x \otimes f \otimes p)| > n_2 \quad (7)$$

where n_2 is a denominator threshold value. The lefthand side of the Eq.(7) is the denominator in the solution for Eq.(4) and (5). The denominator must be large enough to form a reliable result.

The advantage of this method is that it requires only three independent operations: spatial filtering, temporal filtering, and velocity calculation which integrates the filtering results.

3 Object Tracking

First we describe a method for tracking only one object. An object is tracked by updating a rectangular window which circumscribes an object region in the image sequences. The neighborhood of the window is searched for the next object region. This method is not subjected to the influence of the flow vectors which are far from the target object.

Before tracking, we have to determine an object to track in a scene. Here, a first found moving object is tracked. In order to find a moving object, the window size is initialized to the image size, and optical flows are calculated in the window.

The flows that satisfies the next conditions are extracted in the window.

- Inequalities(6) and (7).
- Temporal gradient is sufficiently large. It is represented by next inequalities.
  $g \otimes f \otimes p_t > f_{min}$ and $h \otimes f \otimes p_t > f_{min}$
  where f_{min} denotes threshold.
- The magnitude of the flow vector is not too large. At the pixels where the flow vectors are too large, Eq.(1) is not practical.

When enough flow vectors are obtained, we consider them as belonging to a moving object.

Once the moving object is obtained, it is tracked by the following procedure.

step1 Calculate optical flows in the image.

step2 If the number of pixels with a flow vector in the window is more than a threshold, go to step3. Otherwise, go to step1 to detect a moving object again.

step3 Calculate the mean flow vector in the window.

step4 Search the window and the neighborhood of the window for the pixels whose flow vectors are similar to the mean flow. A set of these pixels is the object region in the next frame.

step5 Update the window so as to circumscribe the object region. If the number of pixels of the region is less than a threshold, the window is not updated.

step6 Return to step1.

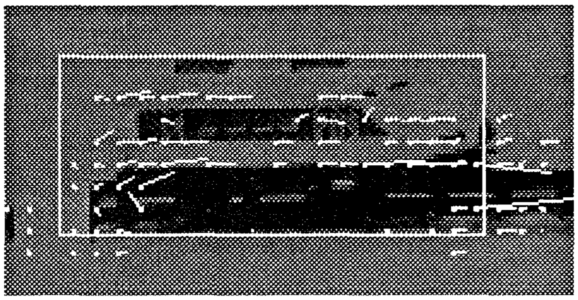
Figure1(a) shows an example of extracted optical flows. The white vectors represent the flow vectors of the toy train. The white rectangle is the window. In Figure1(b), the white regions show the pixels at which the optical flows similar to mean flow are obtained.

This procedure can be applied to tracking multiple objects by multiple windows.

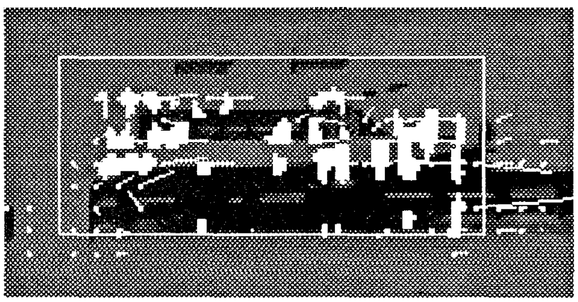
At first, each window size is initialized to image size. If one moving object is found, one window begins to track it. Meanwhile when another object is found, another window begins to track it.

When multiple objects are in the image, some of them may overlap. If two objects overlap, it must be determined which is foreground in step2 and step3 so that windows are correctly updated in step4.

Figure2 shows a situation where windowA and windowB overlap. When two windows are overlapping, each window is divided into two regions: "overlapping region" and "non overlapping region".



(a) Optical flows



(b) Pixels similar flows are extracted

Figure 1: Detecting an object region.

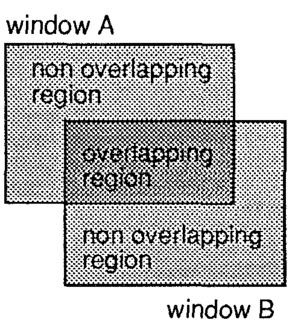
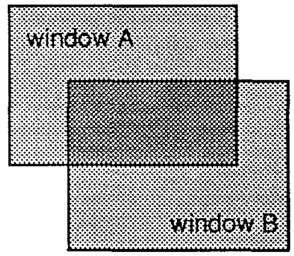
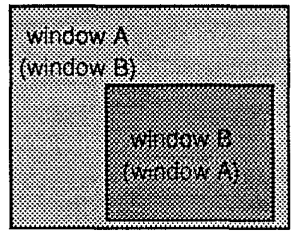


Figure 2: Overlapping windows

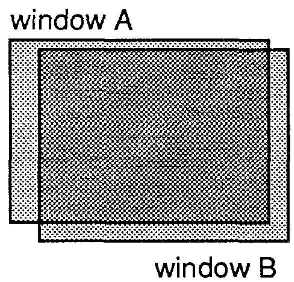
If the number n_o of reliable optical flows in the overlapping region is large enough, the two objects are



(a) case1



(b) case2



(c) case3

Figure 3: Overlapping windows in each case

considered as overlapping. In this case, the decision is made which object is foreground and which is background.

In order to make the decision, let us suppose the following cases according to the number of reliable optical flows (figure3).

Let n_{n_A} and n_{n_B} denote the number of flows in the non overlapping region in windowA and windowB respectively.

case1: $n_{n_A} > n_t$ and $n_{n_B} > n_t$
A part of windowA and a part of windowB are considered to be overlapping.

case2: $n_{n_A} > n_t$ and $n_{n_B} \leq n_t$
or $n_{n_B} > n_t$ and $n_{n_A} \leq n_t$
WindowA (windowB) includes windowB (windowA).

case3: $n_{n_A} \leq n_t$ and $n_{n_B} \leq n_t$
WindowA and windowB are considered to be fully overlapping.

where n_t is a threshold.

For each case, We decide which window is foreground as follows. Let v_o denote the mean flow vector of the overlapping region, and v_{n_A} and v_{n_B} denote the mean flow vector of the non overlapping region of windowA and windowB respectively.

case1:
If $||v_o - v_{n_A}|| < ||v_o - v_{n_B}||$, windowA is the foreground. Otherwise, windowB is the foreground.

case2:
When large window is windowA, if $||v_o - v_{n_B}|| < v_t$(a threshold), windowA is foreground. Otherwise, windowB is foreground.

case3:
The state of each window is the same as the previous frame because the decision can not be made.

In step3 of object tracking, the mean flow vector of the background window is calculated excluding overlapping region. In step4, background window negrects the optical flows in the overlapping region.

Figure4 shows an example of an experiment for two persons crossing in the image.

4 Image Processor

The generalized gradient model is easily implemented on hardware having a linear array architecture. We designed a linear array processor which directly processes TV images in real time. The image processor processes images at 1/15sec interval. Each image consists of 160 $\times$ 120 pixels.

The processor consists of a video I/O board and DSP boards.

The video board digitizes NTSC analog signals from a video camera, and transfers them to the DSP boards. The video board also stores images processed by DSP boards and converts them into analog signals for display.

The DSP board processes digitized images. Each DSP board, shown in Figure5, includes two DSP chips(TMS320C40). These chips are connected in tandem on DSP boards, making up a linear array. In each board, two DSP chips perform independent processings.

The processing in groupA parallels that of groupB. The DSP chip in groupA exchanges data with the other DSP chip in groupB via communication ports.

Optical flow extraction and object tracking are implemented on the image processor as shown in Figure 6.

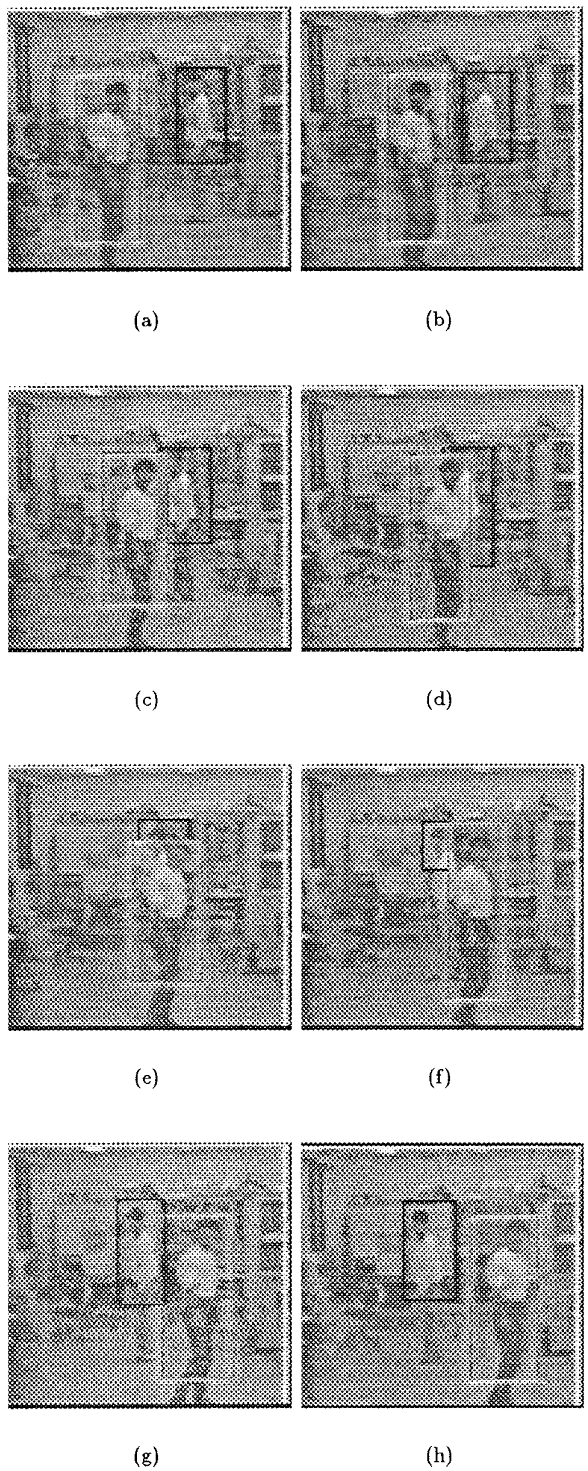


Figure 4: tracking two men

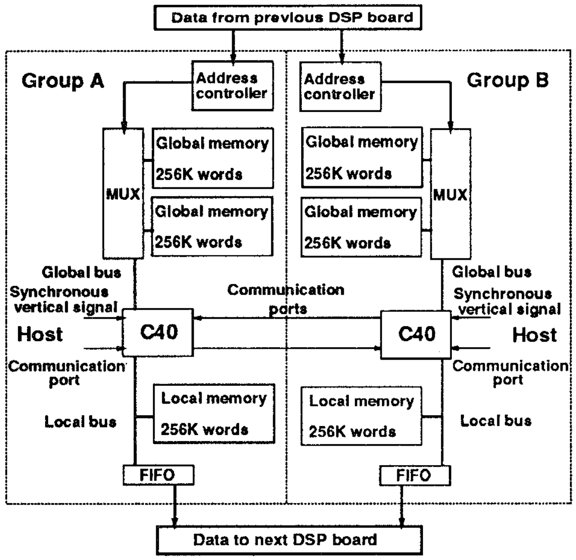


Figure 5: Configuration of the DSP board

Each processing for calculation of optical flow is implemented on the first three DSP boards, spatial filtering on the 1st board, temporal filtering and derivative filtering on the 2nd board, solving equations on the 3rd board.

Object tracking is implemented on the 4th board. The processing for an object is assigned to a DSP. Among two DSP, data of location and scale of the window, and the value of the flow vectors are exchanged via communication port.

The 5th board deals with display processing. Pixels that optical flows are extracted are marked, and windows are drawn. These pixels and windows are colored respectively according to each object.

5 Conclusion

We proposed a method of the object extraction and object tracking based on the optical flow. The method is executed on a special image processor in realtime. The system can track two objects without confusion even if they overlap.

Now the shape of an object is approximated by a rectangular window. If the shape is not rectangular, the precise motion of the object is not obtained. we will improve the method to express the object region more precisely.

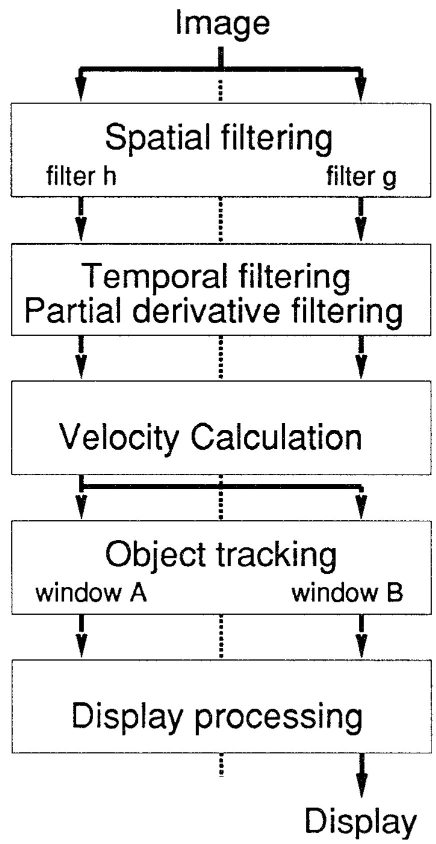


Figure 6: Processing assigned to each DSPboard

In addition, we are planning to extend the system that can use more information such as stereo vision to obtain range data and recognize 3D motion.

Acknowledgments

We would like to express our gratitude to Mr. Sasaki and Mr. Shiohara at Fujitus Laboratory Ltd.

This work is partially supported by the Real World Computing Partnership.

This work was supported by Grant-In-Aid for Scientific Research under Grant No.06555071 from the Ministry of Education, Science, and Culture, Japanese Government.

References

[1] J.L. Barron, D.J. Fleet, S.S. Beauchemin : *Systems and Experiment* Performance of Optical

Flow Techniques, International Journal of Computer Vision, Vol.12, No.1, pp.43-77 (1994)

[2] B.K.P. Horn and B.G. Schunck : Determining optical flow, Artif. Intell., Vol.17, pp.185-203, (1981)

[3] M.V. Srinivasan : Generalized gradient schemes for the measurement of two-dimensional image motion, Biol. Cybern., Vol.63, pp.421-431 (1990)

[4] P. Sobey and M.V. Srinivasan : Measurement of optical flow using a generalized gradient scheme, Journal of the Optical Society of America, pp.1488-1498 (1991).

[5] H. Inoue, T. Tachikawa, and M. Inaba : Robot Vision System with a Correlation Clip for Real-time Tracking, Optical Flow and Depth Map Generation, Proc. IEEE Int. Conf. R&A, vol.2, pp.1621-1626 (1992)

[6] E.D. Dickmanns, R. Behringer, C. Brudigam, D. Dickmanns, F. Thomanek, and V.v. Holt : An All-Tranputer Visual Autobahn-Autopilot/Copilot, Proc. IEEE ICCV'93, pp.608-615 (1993)

[7] M. Brandenbusch, C.K. Sung, and T. Wirtkopp : A vision system for real-time motion analysis in image sequences, FhG IITB Mitteilungen, pp.31-35 (1992)

[8] D.W. Murray, P.F. McLauchlan, I.D. Reid, and P.M. Sharkey : Reactions to Peripheral Image Motion using a Head/Eye Platform, Proc. ICCV'93, pp.403-411 (1993)

[9] M. Shiohara, H. Egawa, S. Sasaki, M. nagle, P. Sobey, M.V. Srinivasan : Real-Time Optical Flow Processor ISHTAR, Proc. ACCV, pp.790-793 (1993).

[10] Y. Shirai, J. Miura, Y. Mae, M. Shiohara, H. Egawa, S. Sasaki : Moving Object Perception and Tracking by Use of DSP, Proc. CAMP, pp.251-256 (1993).