Real-Time Binocular Tracking System

Bo Jia[1], and Noboru Ohnishi[1, 2]

[1] Bio-Mimetic Control Research Center, RIKEN, 2271-130, Anagahora, Shimoshidami, Moriyama-ku, 463-0003, Nagoya, JAPAN
jiab@nagoya.riken.go.jp
[2] Department of Information Engineering, 464-8603, Nagoya, JAPAN
D-69042 Heidelberg, Germany
ohnishi@nuie.nagoya-u.ac.jp

Abstract. We have designed and implemented a real-time binocular tracking system that can robustly track moving objects in a complex environment without any prior knowledge of the object shape or texture. In this system, we use a new motion extraction method that extracts moving objects on the basis of three successive images. We use a new vergence control method based on Two Stage Zero Disparity Filter (TSZDF) to fixate a binocular gaze on an object moving about in a cluttered environment. For real-time motion detection, image subtraction combined with spatial information, using two successive images as a process group, is the most widely used method. This method, however, extracts not only the true motion, but also produces a false motion caused by the occluded background. In order to eliminate the false motion, we present a new motion extraction technique based on three successive images. To maintain the camera's vergence, using a virtual horopter strategy can not only isolate an object from others moving exclusively along the horopter, but also can cope with vergence error caused by the object moving across the horopter. The existing method, however, can only obtain an approximate shift value corresponding to the virtual rotation of cameras. In order to obtain an accurate shift value, we present a new vergence control method based on TSZDF process which involves first the use of coarse ZDF process to localize the coarse-shift value, and then the use of fine ZDF process to obtain the fine-shift value. The two proposed methods are implemented in our real-time binocular tracking system. Experimental results show that the active vision system based on the two proposed methods can effectively track a specified person among many people walking about in a laboratory environment.

1 Introduction

An autonomous humanoid robot [1-2] requires many particular visual functionalities, such as monitoring human actions, learning them by imitation, and helping or protecting them by reacting to observed actions. As a prerequisite for these functionalities, we must develop a real-time active tracking system [3-7], which has the following basic visual skills: the ability to detect and fixate a binocular gaze on a specified moving object, even when there is another moving object in the same visual field. The active tracking system could also function as an automatic cameraman for

many applications such as home video systems, surveillance and security systems, video-telephone systems, television broadcasting systems, or other tasks that are repetitive and tiring for a human being.

In general, there are two different approaches to tracking: 1) recognition-based tracking, and 2) motion-based tracking. The disadvantage of recognition-based tracking is that only the recognizable object can be tracked. On the other hand, motion-based tracking is able to track any moving object without any information about the object such as its size or shape. Therefore, we investigate the motion-based tracking system here.

As a prerequisite for real-time motion-based tracking, we must clearly extract the moving object from successive images robustly. The rotation angles of the camera used for tracking can be controlled according to detected motion. Therefore, the ability to extract motion is a key factor in the development of an active vision tracking system. As was mentioned previously, for practical real-time implementations of motion detection, image subtraction combined with spatial information is the most widely used method [8-10]. In addition to its computational simplicity, this method is suitable for pipeline architecture. Therefore, it can be implemented on most high-speed vision hardware (for example, the MAXVIDEO image processor). On the other hand, if only two successive images are used to extract motion, this method will produce false motion, since background edges that were occluded by the moving object in the first frame only appear in a single frame (namely, the second frame). In order to overcome this disadvantage, we present a new motion extraction technique that uses three successive images as a process group.

For binocular systems, holding the gaze of two cameras is a key problem in terms of tracking a specified moving object among multiple moving objects in cluttered environments. Kaenel [11] presented a Zero Disparity Filter (ZDF) for fixating a binocular gaze on a specified object. The position of the object in the depth direction, however, cannot be estimated from such isolated object. Therefore, we can isolate an object from others using ZDF only if the object is moving exclusively along the same horopter. In order to cope with vergence error caused by the object moving across the horopter, Coombs [12-13] proposed the use of complicated Cepstral filtering technology in order to obtain depth information about the object. This method, however, suffers from the expensive computation necessary for the Cepstral filter. In order to obtain the depth information simultaneously with isolation by ZDF process, Kuniyoshi [14-16] presented an Expanded Zero Disparity Filter (EZDF) based on the concept of the virtual horopter, which significantly decreases the system computation. This method, however, can only obtain an approximate shift value corresponding to the virtual rotation of the cameras. In order to determine an accurate shift value, we present the new vergence control method based on TSZDF process which involves first the use of coarse ZDF process to localize the coarse-shift value, and then the use of fine ZDF process to obtain the fine-shift value. Using the TSZDF process, which is inspired from ZDF and EZDF, we can obtain an accurate shift value with less computation from a broader search range. As a result, we can completely eliminate the vergence error caused by the object moving across the horopter.

An experiment in which we apply our active vision system is performed, and the results demonstrate that the real-time binocular system based on the two new methods can robustly track a specified moving object among multiple moving objects.

In the next section, we first describe the conventional motion extraction method, and then explain the new three-image motion extraction method. In Section 3, we first introduce the ZDF and EZDF methods, and then present the new TSZDF method. In Section 4, we provide the real-time binocular tracking algorithm, and show how to combine the two methods in the binocular tracking system. Section 5 gives experimental results demonstrating the effectiveness of the system. Finally, some conclusions are presented in Section 6.

2 Motion Extraction using Three Successive Images

2.1 Traditional Motion Extraction

Traditional method of motion extraction uses two successive images as a process group. By calculating the temporal derivative of an image and applying a threshold at a suitable level, we can segment an image into regions of motion and inactivity [7]. In general, the temporal derivative can be estimated by simple image subtraction,

$$f_t(x,y;t) = \left| f(x,y;t) - f(x,y;t-d) \right| \quad (1)$$

Techniques for improving image subtraction include spatial edge information to allow the extraction of moving edges. It can be decomposed into two steps:

1) We obtain a binary edge image of the current frame by applying a threshold to the output of an edge detector (here, the Sobel edge detector).

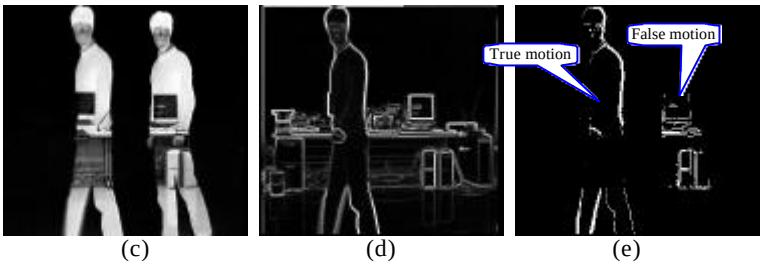
$$f_e(x,y;t) = \sqrt{\left(\frac{\partial f(x,y;t)}{\partial x}\right)^2 + \left(\frac{\partial f(x,y;t)}{\partial y}\right)^2} \quad (2)$$

2) This information is incorporated into the subtracted image by performing a logical AND operation between the two binary images.

$$f_m(x,y;t) = \begin{cases} 1 \text{ if } f_e(x,y;t) \geq T_e \\ \quad \& f_t(x,y;t) \geq T_t \\ 0 \text{ otherwise} \end{cases} \quad (3)$$

The two-step algorithm highlights the edges within the moving region to obtain the moving edges within the second frame. Fig. 1 shows the performance of the traditional method.

(a) (b)



(c) (d) (e)

Fig. 1. Traditional motion extraction using two successive images

In Fig. 1, (a) and (b) are a static successive image sequence, taken at times t-δ and t, respectively; (c) is a subtraction image resulting from equation (1); (d) is an edge image resulting from equation (2); (e) is a moving edge image resulting from equation (3). From (e), it is clear that, although the traditional motion extraction using two images can extract true moving edges, it also produces false moving edges in the area previously occluded by the moving object (since these edges have only been viewed for one sample instant, namely, in the original image taken at time t).

2.2 New Motion Extraction

As shown in Section 2.1, when using a two-image sequence, we cannot extract only the true motion without any false motion, because edges occluded by the moving object only appear in the second frame. On the other hand, these edges also exist in the third frame. Therefore, if three successive images are used as a process group, we are able to eliminate this false motion.

Let us consider the image sequences f(x,y;t-δ), f(x,y;t), and f(x,y;t+δ) taken at times t-δ, t and t+δ, respectively. Let f(x,y;t) be a reference image. We can obtain a moving edge image $f_{m1}(x,y;t)$ from the sequences f(x,y;t-δ) and f(x,y;t) using traditional motion extraction. In a similar manner, we can obtain another one $f_{m2}(x,y;t)$ from f(x,y;t) and f(x,y;t+δ). Finally, we obtain the true moving edge image $f_m(x,y;t)$ by performing a logical AND operation between the two binary images, $f_{m1}(x,y;t)$ and $f_{m2}(x,y;t)$. The above process can be expressed as following:

$$f_{m1}(x,y;t) = T\left[\frac{\partial f(x,y;t)}{\partial t}\cdot\sqrt{\left(\frac{\partial f(x,y;t)}{\partial x}\right)^2+\left(\frac{\partial f(x,y;t)}{\partial y}\right)^2}\right] \quad (4)$$

$$f_{m2}(x,y;t) = T\left[\frac{\partial f(x,y;t+\boldsymbol{d})}{\partial t}\cdot\sqrt{\left(\frac{\partial f(x,y;t)}{\partial x}\right)^2+\left(\frac{\partial f(x,y;t)}{\partial y}\right)^2}\right] \quad (5)$$

$$f_m(x,y;t) = [f_{m1}(x,y;t)]\&[f_{m2}(x,y;t)] \quad (6)$$

where, T(•) denotes the threshold operation. Fig. 2 shows the new motion extraction, using the three-image sequence as a process group.

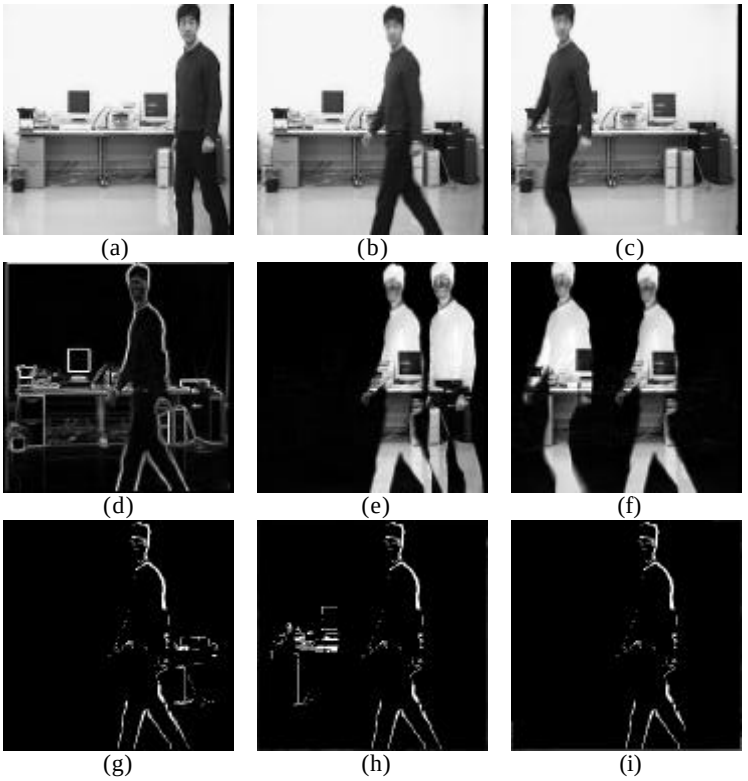


Fig. 2. New motion extraction using three images

In Fig. 2, (a), (b) and (c) are the sequence of images taken at times t-δ, t and t+δ, respectively; (d) is an edge image of reference image (b), resulting from equation (2); (e) is the difference image between (a) and (b), resulting from equation (1); (f) is the difference image between (b) and (c), resulting from equation (1); (g) is a moving-edge image resulting from equation (4), using (d) and (e); (h) is a moving-edge image resulting from equation (5), using (d) and (f); (i) is a moving-edge image resulting from equation (6), using (g) and (h). In (g) and (h), one can see that, the traditional method produces false motion. However, an important point to note here is that the position of false motion differs between (g) and (h), whereas the true motion has the same position. We can retain the true motion and eliminate the false motion easily, as long as we perform a logical AND operation between (g) and (h).

3 Vergence Control Based on Two-Stage ZDF (TSZDF) Process

In binocular systems, whose cameras have their optic axis in the same plane, gaze holding is the process of adjusting the cameras angles so that both are looking at the

same world point. In this section, we first introduce two gaze-holding methods: ZDF and EZDF, and then explain our new method, TSZDF in detail.

3.1 Zero Disparity Filter (ZDF)

We can isolate one object from the others using ZDF process only if the object is moving exclusively along a horopter, which is a set of zero disparity points.

Fig. 3(a) shows the view of binocular fixation in the case where there are two moving objects, C and D. C, the specified object for tracking, is at fixation point of the two cameras. Therefore, its image points have zero disparity. The set of such points is located on a circle passing through the two nodal points of the cameras and the fixation point (here, C). We call this circle as horopter. Thus, C that stays on this horopter can easily be isolated from D that is not on the horopter, by suppressing features with non-zero disparity. This well-known method is called ZDF process. Fig. 3(b) shows the ZDF process result: the disparity of C' is zero; the disparity of D' is d_B (non-zero). Therefore, only C' remains in the output.

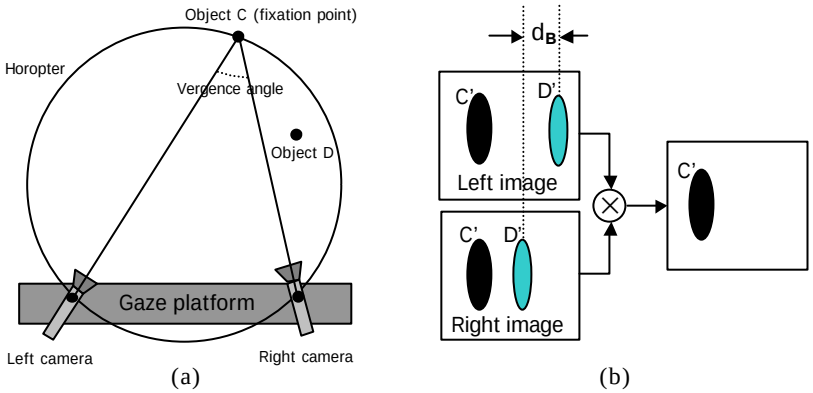


Fig. 3. ZDF process: (a) View of binocular fixation; (b) ZDF process result

3.2 Expanded Zero Disparity Filter (EZDF)

As shown in Fig. 3, the specified moving object C can be isolated from D by ZDF only if C moves along a horopter. In order to cope with vergence error caused by the object moving across the horopter, Kuniyoshi presented an EZDF process based on the concept of a virtual horopter.

Fig. 4(a) shows a view of the virtual horopter, which is a horopter generated by horizontally shifting the right-hand image by a certain amount of pixels, instead of by physically moving the right-hand camera. Low shift values are equivalent to small virtual rotations of the right-hand camera, and vice versa. In this way, we obtain two new virtual fixation points, A_{-S} and A_{+S}, and two new virtual horopters, O_{-S} and O_{+S}, corresponding to the left- and right-hand shift values –S and +S, respectively. Fig. 4(b) shows the matching process in which the pixel numbers of the black areas in the

output denote the matching result. The less pixels there are, the worse the matching result is, namely, the farther the object is from the horopter, and vice versa. In doing so, we have computed three different matching results corresponding to two virtual and one physical position of the horopter. As a result, the object may be located on the horopter that produces the best matching (here, the virtual horopter O_{-S}).

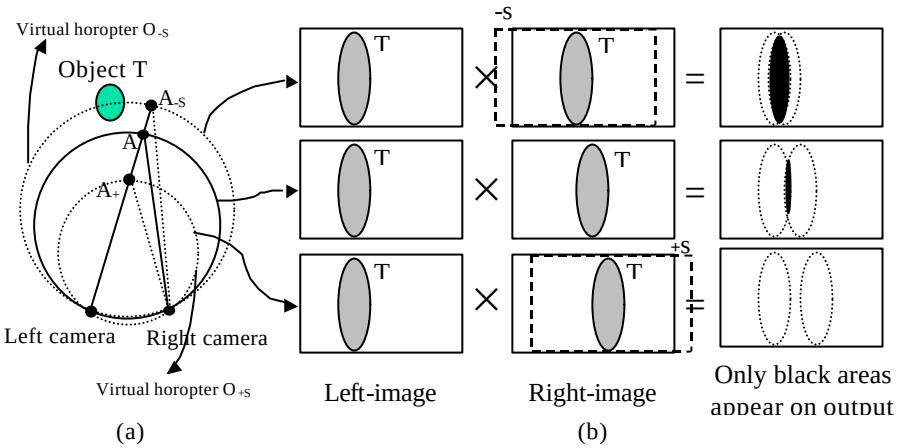


Fig. 4. EZDF process: (a) View of virtual horopter; (b) EZDF process result

3.3 Two-Stage Zero Disparity Filter (TSZDF)

As shown in Section 3.2, EZDF can cope with vergence error, and obtaining a suitable shift value is a key problem for fixating a gaze on an object moving across a horopter. The EZDF method uses equation (7) to obtain the shift value S,

$$S = w_{\max} \times (1 - N / N_{\max}) \tag{7}$$

here, ω_{max} is the width of the edge, N is the practical pixel number of ZDF output, and N_{max} is the pixel number of ZDF output in the case of no vergence error. The evaluation of N_{max} is approximate, since we can obtain N_{max} only if we eliminate the vergence error, and this is what we intend to solve as an end product of EZDF.

The EZDF method can only yield an approximate shift value, which directly affects the effectiveness of maintaining the vergence. In order to obtain an accurate shift value, we present the new vergence control method.

3.3.1 Coarse ZDF Stage

Imagine that there is an object A moving across a horopter. Suppose that A_1 and A_2 are its projections onto the left- and right-hand images, respectively, with a width of 1 pixel (shown in Fig. 5). The difference between the positions of A_1 and A_2 is 7 pixels, which is caused by vergence error. Assume that the search range of the shift value for eliminating the vergence error is [-10, 10].

We first perform a dilation operation for images f_1 and f_2. Thus, we obtain images g_1 and g_2, in which the widths of B_1 and B_2 are 3 pixels, respectively. Then, we select the coarse-shift step of 5 pixels (referring to Section 3.3.3), and compare the corresponding matching results. In this way, we computed five different matching results corresponding to the coarse-shift values of -10, –5, 0, 5 and 10. As a result, we can localize the coarse-shift as –10, which produces the best matching in the ZDF output (here, the width of its matching output is 1 pixel).

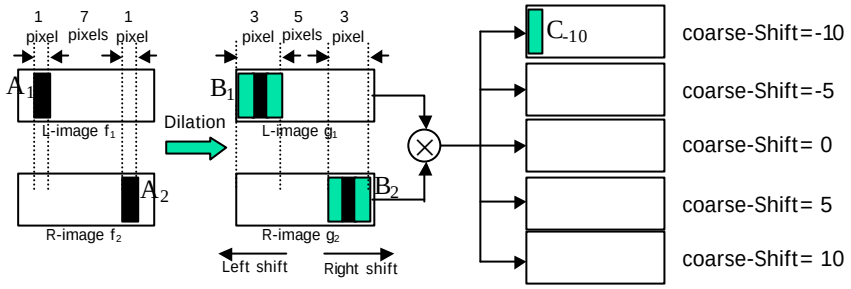


Fig. 5. Conceptual example of the Coarse ZDF process

3.3.2 Fine ZDF Stage

Suppose that we obtained the coarse-shift of –10 by Coarse ZDF. Now we discuss how to obtain the fine-shift by Fine ZDF.

We first obtain image g_3 by horizontally shifting image g_2 by –10 pixels (shown in Fig. 6). Next, we select the fine-shift step of 1 pixel (referring to Section 3.3.3). Then, we obtain two different matching results corresponding to the fine-shift of –1 and 1. Next, we compare them depending on the number of matching pixels in the output, and determine the true direction of shift (here, the true direction of shift is to the right). As a result, we obtain the third matching result corresponding to the fine-shift of 2. Finally, we can obtain the fine-shift as 2, which produces the best matching in the ZDF output (here, the width of its matching output is 3 pixels). Therefore, the final-shift is the sum of the coarse-shift and fine-shift (namely, –8 pixels).

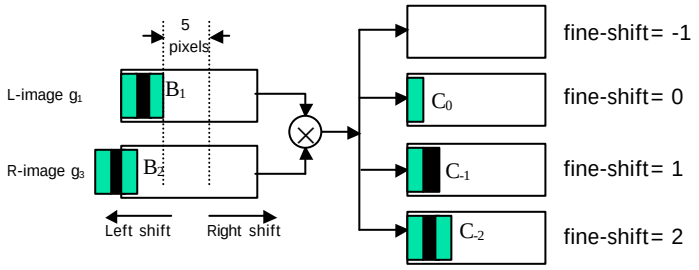


Fig. 6. Conceptual example of the Fine ZDF process

3.3.3 Parameters Used in TSZDF

Suppose that the width of the moving edge is W pixels, and the search range of the shift value is [-R, R] pixels. We can calculate the coarse-shift step S_R and the ZDF number N_R in the Coarse-ZDF stage, using equations (8) and (9), respectively

$$S_R = 2 * W - 1 \tag{8}$$

$$N_R = Int((2 * R + 1) / S_R) + 1 \tag{9}$$

here, Int[•] denotes the integration operation. As presented in Section 3.3.1, W is 3 pixels and R is 10. Therefore, S_R is 5 pixels, and N_R is 5.

Let the fine-shift step be S_F. We can calculate the ZDF number N_F in the Fine-ZDF stage, using equation (10)

$$N_F = Int((W - 1) / S_F) + 1 \tag{10}$$

As presented in Section 3.3.2, W is 3 pixels and S_F is 1 pixel. Therefore, the N_F is 3.

The calculation number of ZDF in TSZDF process is the sum of N_R and N_F.

4 Tracking Algorithm

In Sections 2 and 3, we presented two new methods. Now, we discuss their effective combination in the real-time binocular tracking system.

4.1 Tracking Algorithm

Fig. 7 shows the tracking algorithm, which can be decomposed into four parts:

(1) Motion extraction: used to extract the moving objects in both images.
(2) Coarse ZDF: used to check whether the two cameras approximately converge at the specified moving object.
(3) Saccade: used to modify the angles of the two cameras, in order to cause them to converge approximately at the moving object.
(4) Fine ZDF: used to obtain the accurate shift value, in order to completely eliminate vergence error and exactly maintain vergence.

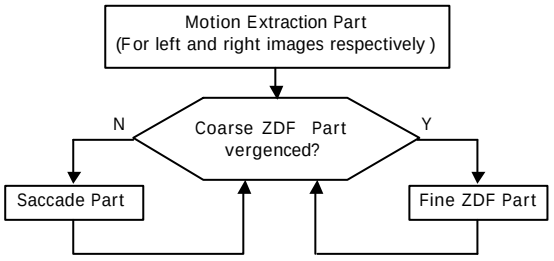


Fig. 7. Tracking algorithm

4.2 Motion Extraction Part

We must calculate the following information needed for subsequent parts, once the moving edges have been extracted (refer to Section 2).

4.2.1 Calculation of Position of Moving Object

We can obtain the position information for the moving object using equation (11).

$$\begin{cases} P_{x,t} = \dfrac{\sum x f_m(x, y; t)}{\sum f_m(x, y; t)} \\ P_{y,t} = \dfrac{\sum y f_m(x, y; t)}{\sum f_m(x, y; t)} \end{cases} \tag{11}$$

here, ($P_{x,,t}$, $P_{y,,t}$) denotes the centroid coordinate of the moving object at time t.

4.2.2 Calculation of Rotation of Cameras

Let us consider a perspective projection coordinate frame as shown in Fig. 8.

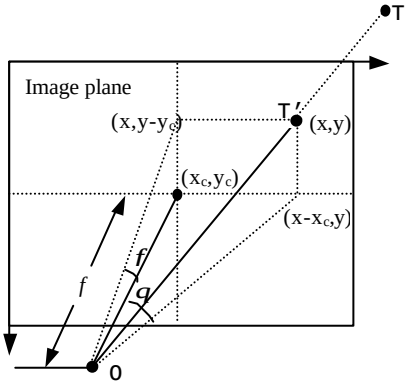


Fig. 8. Camera geometry with perspective projection

In Fig. 8, T is an object point in the 3D world system, and (x,y) is its image coordinate. From Fig. 8, we can obtain equation (12), which transforms the planar coordinate ($P_{x,t}$, $P_{y,t}$) of T to the spherical coordinate (è(t), ö(t)) in time t.

$$\begin{cases} \theta(t) = \tan^{-1}\left[\dfrac{d_x}{f} \cdot (P_{x,t} - x_c)\right] \\ \phi(t) = \tan^{-1}\left[\dfrac{d_y}{f} \cdot (P_{y,t} - y_c)\right] \end{cases} \tag{12}$$

here, d_x and d_y are center-to-center distances between adjacent sensor elements in X and Y direction respectively, f is effective focal of camera.

4.3 Coarse ZDF and Saccade Part

In this section, we use an example (shown in Fig. 9) to explain how to cause two cameras to converge approximately at a moving object using the Coarse ZDF and Saccade processes.

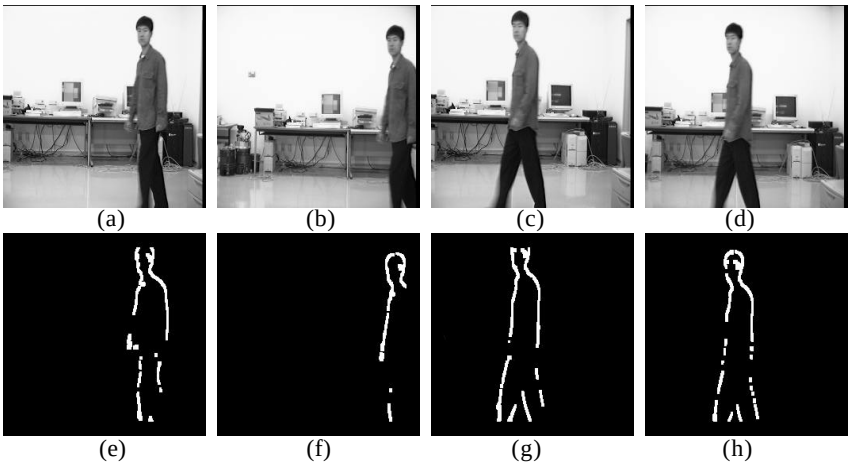


Fig. 9. The Coarse ZDF and Saccade process

In Fig. 9, (a) and (b) are a pair of left- and right-hand images taken at time t-δ; (e) and (f) are the corresponding dilated moving edge images of (a) and (b). After the Coarse ZDF and Saccade processes, we obtain another pair of left- and right-hand images, (c) and (d), taken at time t; (g) and (h) are the corresponding dilated moving edge images of (c) and (d). In this experiment, we assume that, R is 10 pixels, and W is 3 pixels. From (e) and (f), one can see that the difference in the position of the walking person between (a) and (b) is very large (here, it is 47 pixels, which is larger than R). As a result, the ZDF outputs corresponding to different coarse-shifts are all zero in Coarse-ZDF process. Therefore, the algorithm determines that the two cameras do not converge at the walking person, and we move on to the Saccade process. In the Saccade process, we first obtain the current position information for the walking person in (a) and (b), using equation (11). Then, we obtain the rotation angles of the left- and right-hand cameras, using equation (12). Finally, we drive the motor to rotate the two cameras. After this, we obtain (c) and (d). From (g) and (h), it is clear that the two cameras converge approximately at the walking person.

4.4 Coarse ZDF and Fine ZDF Part

In this section, we use another example (shown in Fig. 10) to explain how to completely eliminate vergence error and exactly maintain vergence using the Coarse ZDF and Fine ZDF processes.

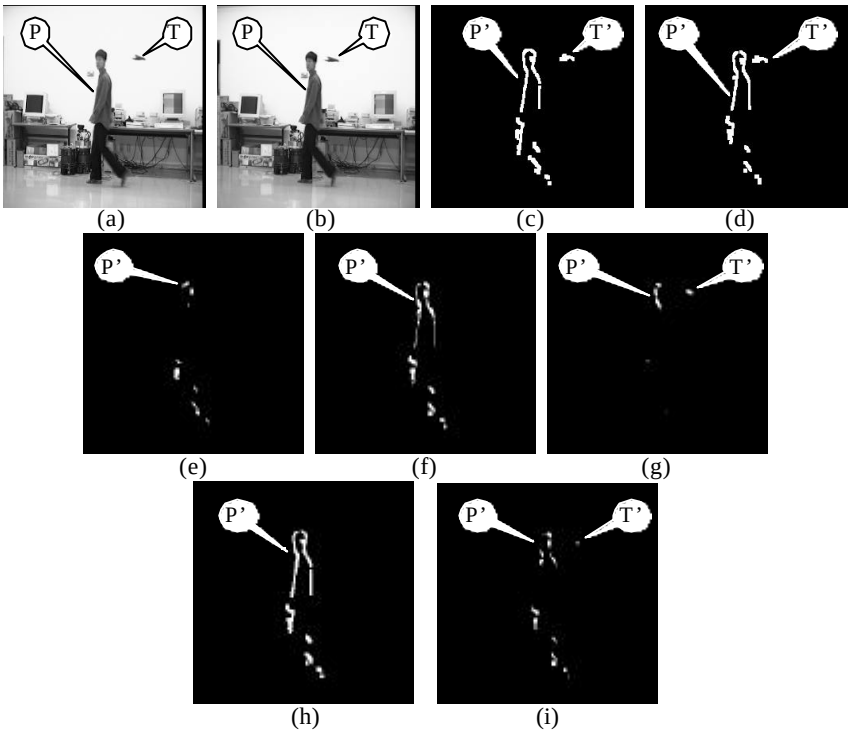


Fig. 10. The Coarse ZDF and Fine ZDF process

In Fig. 10, (a) and (b) are a pair of left- and right-hand images taken at time t-δ; (c) and (d) are the corresponding dilated moving edge images of (a) and (b). In this experiment, we assume that R is 14 pixels, W is 4 pixels and S_F is 2 pixels. Therefore, S_R is 7 pixels. Using the Coarse ZDF process, we obtain three non-zero ZDF output images, (e), (f) and (g), corresponding to coarset-shifts of 0, 7 and 14 pixels, respectively. As a result, the algorithm reveals that the two cameras converge approximately at the walking person, and the coarse-shift is 7 pixels (since (f) is a better matching than (e) or (g)). The algorithm will move on to the Fine ZDF process. Using the Fine ZDF process, we obtain another two non-zero ZDF output images, (h) and (i), corresponding to fine-shifts of –2 and 2 pixels, respectively, in which (h) is a better matching than (i). Therefore, the algorithm yields the fine-shift of –2 pixels, and the final-shift, which is the sum of the coarse-shift and fine-shift, is 5 pixels. In fact, the difference in the position of the walking person between (c) and (d) is 4 pixels. The reason why there is a one-pixel precision error in this experiment is that we previously set the fine-shift step S_F as 2 pixels; if we had set it as 1 pixel, then there would be no precision error (However, this would increase the computation).

In (a) and (b), P is a person walking about in the laboratory, and T is a toy aircraft flying around in the laboratory. In (c) to (i), P' is the dilated moving edges of P, and T' is the dilated moving edges of T. In (h), one can see that only P' remains in the output. The reason is that the algorithm completely eliminates vergence error by using

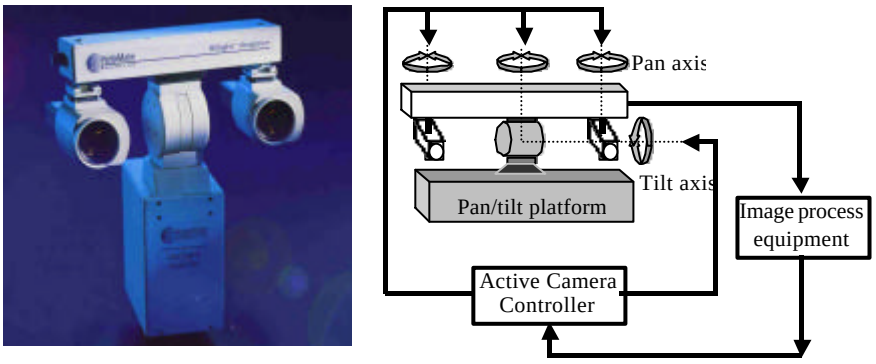
the Coarse ZDF and Fine ZDF processes; therefore, it can keep the two cameras exactly converging at the walking person.

5 Experiment

In this section we present experimental results for the real-time binocular tracking system based on the new motion extraction and vergence control methods. The parameters used in the experiment are the image size of 256*242; R of 14 pixels; W of 4 pixels; S_F of 2 pixels; and S_R of 7 pixels.

5.1 System Structure

The system consists of two cameras, all-purpose image processing equipment, an active camera controller, and a motor-driven pan/tilt platform. Fig. 11 shows the system structure. There are two cameras on the pan/tilt platform (HelpMate Robotics Inc. Unisight Pan/Tilt platform) to capture images of a scene. The input images are then processed by the all-purpose image processing equipment, which consists of a host computer (MVME167, LynxOS), an image processing VEM board (Datacube MAXVIDEO 250) and a full-color image in-out board (Datacube, DIGICOLOR); both of boards are inserted into the VEM slots of the host computer. The pan/tilt platform is controlled by an active camera controller (Delta Tau Data System Inc. PMAC motion controller) for the rotation about the pan and tilt axes, and for the rotation of the cameras. The camera controller receives the commands sent by the host computer, interprets them, and generates the corresponding controlled pulsed to drive the motor of the pan/tilt platform.



(a) (b)

Fig. 11. System structure: (a) the Bisight; (b) the system structure

5.2 Pipeline Process

In order to explain how to realize the algorithm on the image process VEM board MAXVIDEO 250, we describe the pipeline process of the algorithm for the case of convergence (shown in Fig. 12).

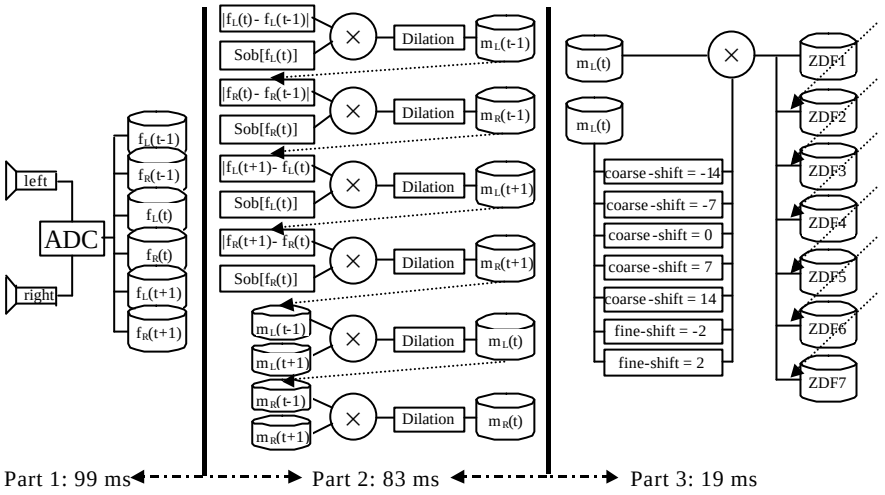


Fig. 12. Pipeline process of the algorithm (in the case of convergence)

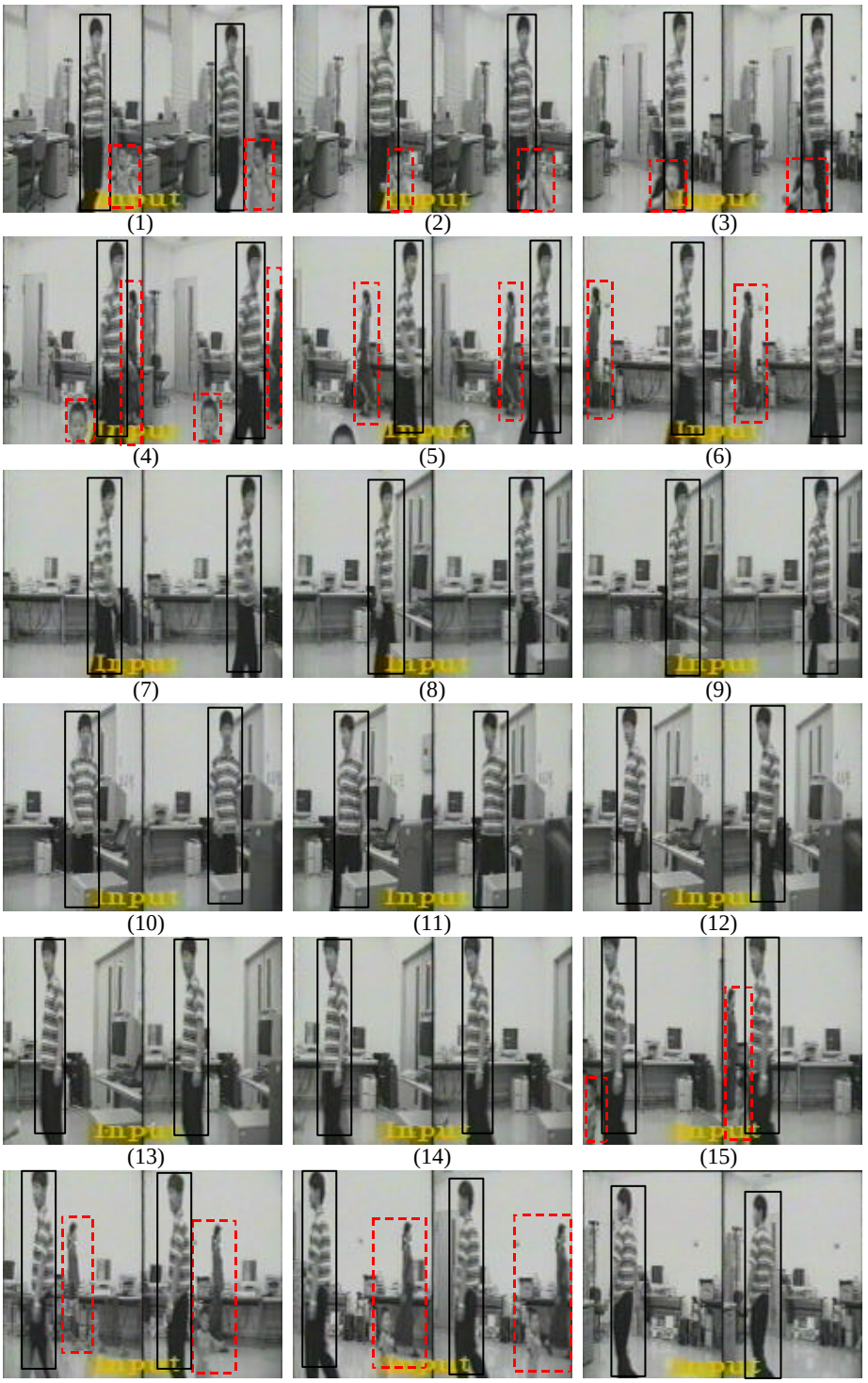
In Fig. 12, a rectangle denotes a process; a cylinder denotes a memory block that holds a specified image; $f_L(t\text{-}n)$ denotes an image taken by the left-hand camera at time (t-n); Sob( ) denotes a sobel difference operation; $m_L(t\text{-}1)$ denotes a moving edge image obtained from $f_L(t\text{-}1)$ and $f_L(t)$; $m_L(t+1)$ denotes a moving edge image obtained from $f_L(t+1)$ and $f_L(t)$; $m_L(t)$ denotes a moving edge image obtained from $f_L(t\text{-}1)$, $f_L(t)$ and $f_L(t+1)$; and the broken line with an arrowhead denotes the process sequence of the pipeline. From Fig. 12, it is clear that the pipeline process can be decomposed into three parts in the case of convergence:

(1) Part 1: reading three successive images used in part 2. This will take 99ms.
(2) Part 2: extracting moving objects used in part 3. This will take 83ms.
(3) Part 3: performing the TSZDF process. This will take 19ms, in which case each ZDF process requires 2ms.

The time used for other processes, such as communication between the host computer and the camera controller, and the rotation of cameras driven by motors, is about 247ms. The total time required for the tracking algorithm is about 448ms.

5.3 Experimental Results

Here, we present a real-time tracking experiment that was conducted in a laboratory environment. Fig. 13 shows 21 frames obtained in the experiment.


(1)
(2)
(3)
(4)
(5)
(6)
(7)
(8)
(9)
(10)
(11)
(12)
(13)
(14)
(15)

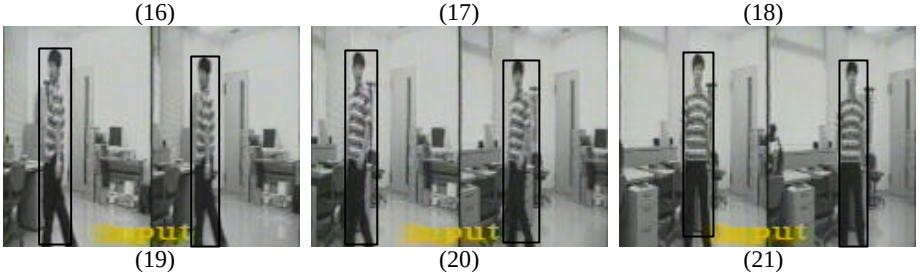


Fig. 13. Real-time binocular tracking results: frame interval is 0.5s

In this experiment, there are three objects walking simultaneously in the scene: a man, a woman and a child. In (1), the two cameras initially converged at the man. Therefore, the two cameras can maintain convergence at the man as he walks through the laboratory, even through the woman and child also walk into the field of vision of the cameras. Fig. 13 shows that the real-time binocular system can track a specified person walking about in a complex environment robustly. It should be noted that, the walking man is not always at the center of the image sequence, since there is some tracking delay due to computational cost. There are two solutions: one is to improve the algorithm in order to decrease computational time; another is to consider the velocity in order to forecast the next possible position.

6 Conclusion

In this paper, we introduced a real-time binocular tracking system, which includes two new methods:

1) The new motion extraction method based on the use of three successive images: it not only obtains the true motion, but also effectively eliminates false motion caused by the occluded background, which is a disadvantage of the traditional motion extraction method.
2) The new vergence control method based on the TSZDF process: EZDF is an effective method for coping with vergence error. This method, however, can only yield an approximate shift value corresponding to the virtual rotation of the cameras. Therefore, we present a TSZDF, which can yield an accurate shift value with less computational cost. That is, it completely eliminates the vergence error caused by the object moving across the horopter.

The two new methods have been tested in combination with a real-time binocular tracking system. Experimental results show that the real-time binocular system can robustly track a specified person among many people walking about in a complex environment.

References

1. Kuniyoshi, Y.: Humanoid as Research Vehicle into Flexible Complex Interaction. Proc. Int. Conf. on Intelligent robots and systems (IROS'97), Grenoble, France, Sep., 1997
2. Kuniyoshi, Y., Rougeaux, S.: Humanoid Vision System for Interactive Robots. Proc. of The First Asian Symposium on Industrial Automation and Robotics, Bangkok, Thailand, May, 1999, pp.13-21
3. Ballard, D.H., Brown, C.M.: Principles of Animate Vision. CVGIP: Image Understanding, 1992
4. Christensen, H.I., Bowyer, K.W., Bunke, H.: Active Robot Vision. World Scientific, 1993
5. Takeuchi, Y., Wang, Z. F., Ohnishi, N., Sugie, N.: Real-Time Visual Tracking System Suggested by Saccade and Pursuit Eye Movements. Journal of the Robotics Society of Japan, Vol. 15, 1997, pp.474-480
6. Rougeaux, S., Kuniyoshi, Y.: Robust Real-Time Tracking on an Active Vision Head. Proc. Int. Conf. on Intelligent robots and systems (IROS'97), Grenoble, France, Sep., 1997
7. Scassellati, B.: A Binocular, Foveated, Active Vision System. Technical report, Massachussetts Institute of Technology, January, 1998
8. Murray, D., Basu, A.: Motion Tracking with an Active Camera. IEEE Trans. on PAMI., Vol. 16, 1994, pp. 449-459.
9. Picton, P.D.: Tracking and Segmentation of Moving Objects in a Scene. Proc. 3rd Int. Conf. Image Processing and its Applicat., Coventry, England, 1989, pp. 389-393
10. Allen, P., Yoshimi, B., Timcenko, A.: Real-time Visual Servoing. Tech. Rep. CUCS-035-90, Columbia Univ., Pittsburgh, PA, Sept. 1990
11. Kaenel P.V., Brown, C.M., Coombs, D.J.: Detecting Regions of Zero Disparity in Binocular Images. Tech. Rep. of University of Rochester, TR388, 1991
12. Coombs, D.J.: Real-Time Gaze Holding in Binocular Robot Vision. Tech. Rep. Of University of Rochester, TR415, 1992
13. Coombs, D.J., S., Brown, C.M.: Real-Time Smooth Pursuit Tracking for a Moving Binocular Robot. Computer Vision and Pattern Recognition, 1992, pp. 23-28
14. Kuniyoshi, Y.: A Compact Mobile Robot with Binocular Tracking Vision. Journal of the Robotics Society of Japan, Vol. 13, 1995, pp. 343-346
15. Kita, N., Rougeaux, S., Kuniyoshi, Y., Sakane, S.: Real-Time Binocular Tracking Based on Virtual Horopter. Journal of the Robotics Society of Japan, Vol. 13, 1995, pp. 683-690
16. Rougeaux, S., Kita, N., Kuniyoshi, Y., Sakane, S.: Tracking a Moving Object with a Stereo Camera Head. The 11th Annual Conf. Of Robotics Society of Japan, 1993, pp. 373-376