MVA2000 IAPR Workshop on Machine Vision Applications, Nov. 28–30, 2000, The University of Tokyo, Japan

1—4 Automatic Intruder Detection and Tracking System with Pan-Tilt-Zoom Camera

Tetsuji Haga * Hideto Fujiwara * Kazuhiko Sumi *
Industrial Electronics and Systems Lab., Mitsubishi Electric Corporation

Abstract

We propose an automatic monitoring system which detects and tracks intruders, using a pan-tilt-zoom camera. Detection is based on a background subtraction method which detects the difference between input image and continuously updated background images. Tracking is based on a newly developed algorithm incorporating two complementary methods: template matching and frame difference. When both methods are available, the template position is corrected with the frame difference data, enabling the robust tracking of intruders regardless of shape, aspect, and moving velocity. When either of the two methods is not available, the other method continues tracking alone till both methods are available again. Tracking mode is selected automatically according to prevailing conditions.

1 Introduction

In water or power plants, automatic video surveillance systems is necessary, to prevent accidents and subversive activity. Two functions are required: automatic intruder detection, to sound the alarm; and intruder tracking with the pan-tilt-zoom control, to keep intruders in sight of the camera and record details of their characteristics and behavior.

For intruder detection, the background subtraction method [1, 2, 3, 4, 5] is available for use even in outdoor conditions. This method detects the intensity difference between the input image and a background image that is continuously renewed, according to illumination variation and the noise model. However, the background subtraction method is not available for intruder tracking, because once the camera changes position or zoom magnification, background image renewal is impossible.

The frame difference method detects moving objects by comparing two neighboring frame images. This is effective while the camera is still, and background change, other than the moving object, is small.

The optical flow method [6, 7, 8] detects moving objects by gathering the similar motions calculated between between two neighboring frame images. This method is available while the camera is in motion. However, for human tracking, region growing based on motion is very difficult, because the motion vectors in non-rigid and deformable objects, like human bodies are irregular in direction. Moreover, motion might be very small, even unreliable, depending on geometric conditions and zoom magnification.

The template matching method [9] detects moving objects by searching the best matched block pattern with the template. This is also available while the camera is in motion. However, the position gap between the template and tracking object grows significant as time passes, and the system can eventually lose the object. Moreover, when the tracking object is partially obscured, matching reliability can descend.

We propose a new tracking algorithm, which utilizes both template matching and frame difference, to bring out the respective advantages of these complementary methods, and effect more reliable tracking. The size and location of the tracking object, extracted by the frame difference, are used to cancel out the difference between the template location and that of the tracking object, and used to adjust the zoom magnification. Moreover even if one of the two methods is not fully available, the other method can continue tracking alone.

In this paper, we describe the basic principles of our proposed intruder detection and tracking system, as well as the results of experiments with actual images.

2 Proposed Method

2.1 Outline of the Intruder Detection and Tracking Process

Figure 1 shows the processing flow of the proposed intruder detection and tracking system. In the intruder detection mode, the system processes the background subtraction to determine the region where significant intensity changes have occurred. If the area does not exceed the pre-defined threshold, the system does not sense an intruder, and continues updating background images. On the other hand, if the area is larger than the threshold, the system determines there is an intruder. In this case, the

*Address: 8-1-1, Tsukaguchi Honmachi, Amagasaki, Hyogo, 661-8661 Japan. E-mail: haga@img.sdl.melco.co.jp

system sets up the template pattern to includes the region the significant intensity change was detected, and switches to tracking mode.

If the system loses track of the intruders and cannot rediscover him/her within a certain time, the system moves the camera to the initial position, and switches back to intruder detection mode.

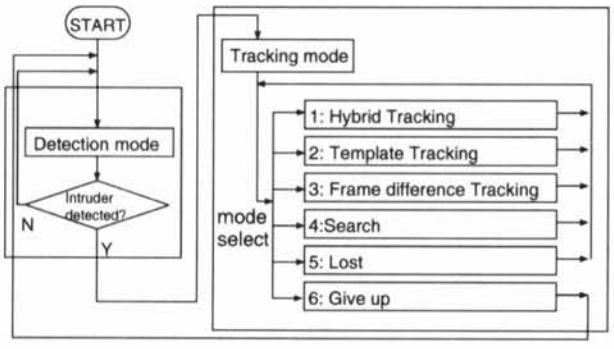

Figure 1: Processing flow

2.2 Details of the Tracking Algorithm

In the tracking mode, the system tracks the intruder and sends the camera pan, tilt, and zoom commands, to keep the intruder inside the camera frame.

2.2.1 Hybrid Tracking Mode (*TM+FD*)

When both the reliability of the template matching and frame difference methods are sufficiently reliable, the system combines them, i.e. hybrid tracking(Figure 2).

For our purposes, the tracking object is a human body, with complex changes in aspect and shape. So it is preferable to set the template position on an intruder's head or upper half body, which make smaller deformations than the lower half.

In the beginning of the tracking mode, the system sets a template of size $wx \times wy$, including the intruder's head or upper half body. Then, matching is done to obtain the tracking position (x_t, y_t), and the motion from the previous frame (u, v) (Figure 2(a)).

From the next processing cycles, the template is set up again, according to the best matched position (x_t, y_t). If the tracking is done by template matching alone, slight differences accumulate between the template position and that of the object, as the template is updated, until the template finally loses the object. Using the object's position and size, as obtained by the frame difference, corrects template position so as not to lose the tracking object.

Frame difference is calculated in the limited region around (x_t, y_t), i.e. the best matched position, to obtain the gravity point (x_g, y_g), and the circumscribed rectangle (sx, sy, ex, ey). Here, (x_g, y_g) is the average gravity point of the object, before and after the motion. The gravity point of the object after the motion (x_f, y_f) is obtained using the motion vector (u, v) obtained by template matching (Figure 2(b)).

$$\begin{cases} x_f &= x_g + u/2 \\ y_f &= y_g + v/2 \end{cases} \quad (1)$$

(sx, sy, ex, ey) is the circumscribed rectangle of the region where the object exists, before and after the motion. These coordinates should also be corrected using the motion vector (u, v) obtained by template matching (Figure 2(c)).

$$\begin{cases} if(u > 0) & : & sx = sx + u \\ if(u < 0) & : & ex = ex + u \\ if(v > 0) & : & sy = sy + v \\ if(v < 0) & : & ey = ey + v \end{cases} \quad (2)$$

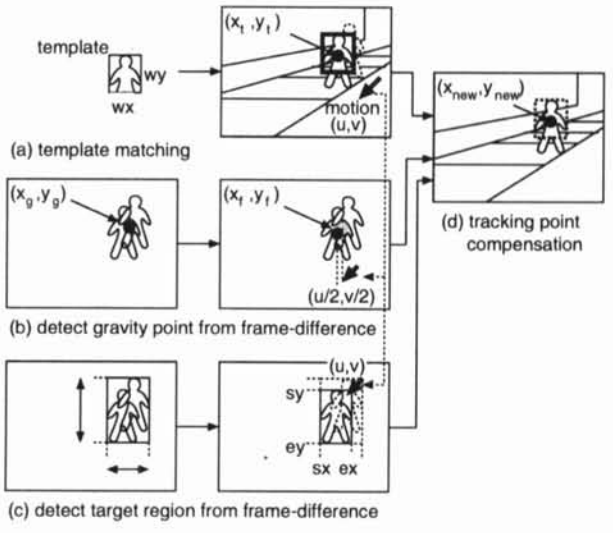

Figure 2: Principles of the hybrid tracking

To set the template in a position where the the upper side of the template and that of the intruder are equivalent, the x coordinate of the gravity point x_f is used as it is for the horizontal position. For the vertical position, y_f is transformed as follows:

$$\begin{cases} x_f &= x_g + u/2 \\ y_f &= sy + wy/2 \end{cases} \quad (3)$$

The ultimate correction of the tracking point (Figure 2(d)) is applied according to (x_{ofst}, x_{ofst}), the offset of the tracking position obtained by template matching (x_t, x_t) and that obtained by frame difference (x_f, y_f). When each coordinate of the offsets x_{ofst} or x_{ofst} is smaller than the pre-defined limit x_{lim}, y_{lim}, the offset x_{ofst}, x_{ofst} are used as they are. In case the offset is larger than the limit, the limit is substituted with the offset.

$$\begin{cases} x_{ofst} &= \begin{cases} x_f - x_t & (x_f - x_t \leq x_{lim}) \\ x_{lim} & (otherwise) \end{cases} \\ y_{ofst} &= \begin{cases} y_f - y_t & (y_f - yt \leq y_{lim}) \\ y_{lim} & (otherwise) \end{cases} \end{cases} \quad (4)$$

The final setup position tracking point is obtained by adding the offset (x_{ofst}, x_{ofst}) to the position (x_t, x_t) :

$$\begin{cases} x_{new} = x_{tm} + x_{ofst} \\ y_{new} = y_{tm} + y_{ofst} \end{cases} \quad (5)$$

Thus, the template drift in one direction is eliminated, enabling reliable tracking of an object. As the tracking object is not rigid, and its shape deforms between frames, (x_f, y_f) does not represent the precise gravity point of the object. Nor does (x_{ofst}, y_{ofst}) represent the exact difference between the template position and that of the object. Still, using the limit (x_{lim}, y_{lim}), the system provides satisfactory results, despite fluctuations by some pixels. The limit values are a pixel or two, in both directions.

2.2.2 Template Tracking Mode(*TM only*)

While the camera is panning or zooming, or still swaying right after some camera control, the frame difference method is not available. In such cases, the system tracks the object by template matching alone. The size of the object is taken over from the previous frame as it is. If tracking continues in this mode over a long frame, small differences between the template and the object might accumulate enough for the system to lose sight of the object. To prevent this, new camera commands are inhibited in this mode.

2.2.3 Frame Difference Tracking mode(*FD only*)

When part of the object being tracked is concealed by another object, the system can still track the object reliably using frame differences alone, as long as the camera is stable. In such a case, template matching cannot process the object's motion, resulting in some errors. Still, such errors can be safely ignored in actual tracking, since the object's moving vector is only several pixels in length, except for rare occasions when the intruder runs at his/her full speed in a direction orthogonal to the camera's line of vision.

2.2.4 Sub-modes

In addition to the modes described so far, we have sub-modes to resume the tracking when the system loses sight of the object. Even after the object stands still or hides itself behind another object, if frame differences show an area where changes were detected longer than a defined period of time, the system jumps the point to track to a new position. This is the frame difference search mode (*SEARCH*). If no such area is detected, the system waits at the previous point to track. This is the waiting mode for temporarily lost objects (*LOST*). If the camera moves while template matching does not work, the system returns the camera to its home position and uses initial background difference. This is the return to the initial state mode (*GIVE UP*).

2.3 Tracking Sub-mode Selection

The system chooses the tracking sub-modes depending on the case distinctions listed in Figure 3. In making this selection, the system obtains the following four indexes from the frame differences and template matching:
(1) Whether template matching is effective or not;
(2) Whether the camera is stable or vibrating;
(3) Whether a frame difference exists in the locality around the point to track; and
(4) Whether a frame difference exists in the scene (except the point to track).

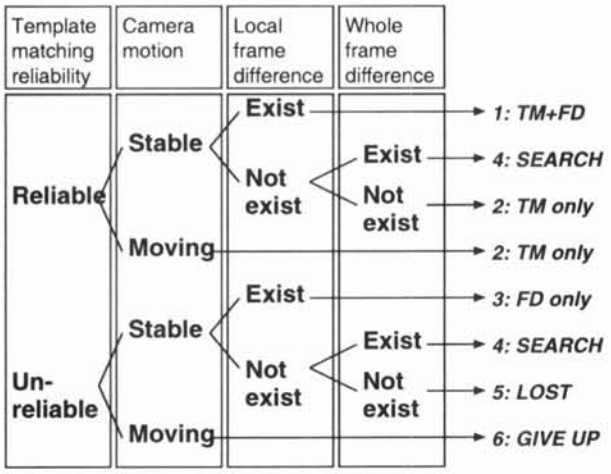

Figure 3: Selection of the tracking sub-modes

Figure 4 outlines how the system decides whether template matching is effective or not. The system reads the score data obtained by template matching, then computes the difference between the peak value and the average value of the eight points separated from the peak point by $\pm w/2$ in both x and y directions. The system also determines the peak gradient by dividing d by w, then compares this gradient with the threshold value. For example, Figure 4 (a) shows a steep peak, with which template matching works. Figure 4 (b) shows a gentle peak, in which case template matching is deemed to be ineffective.

Figure 5 illustrates how the system determines whether the camera is stable or vibrating. The system computes the number of edge segments from the

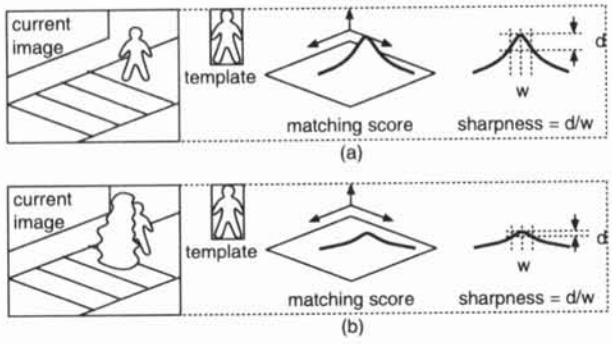

Figure 4: Determining the reliability of template matching

frame difference data, and compares the result with that from the spatial differential edge computation. If this comparison shows a large difference, as in (a), the camera is considered to be stable. If there is little difference, as in (b), the camera is determined to be vibrating.

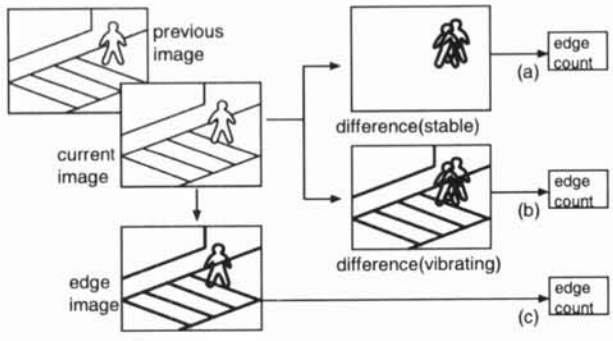

Figure 5: Determining the stability of the camera

Figure 6 shows how the system determines whether frame difference exists in the locality around the point to track. To do so, the system computes the edge within the limited locality and compares this value with the threshold value obtained from past results.

Figure 6 also shows how the system determines whether a frame difference exists in the whole scene. To do so, the system computes the edge from the frame difference data in the entire scene, and compares this value with the threshold value obtained from past results.

2.4 Camera Control

Our proposed camera is manipulated only when the object being tracked is about to go outside the camera frame, i.e., the object is going out of the area (CC) in Figure 7 (a). When this occurs, the camera pans or tilts in a pre-defined direction for each corresponding area. The camera zooms only

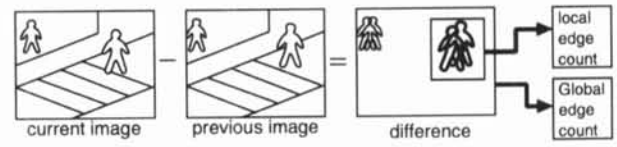

Figure 6: Determining whether frame difference exists

when the image of the object being tracked is about to exceed the designated size, i.e., when the height of the object h is about to grow larger than 1/5 of that of the image ly, in Figure 7 (b).

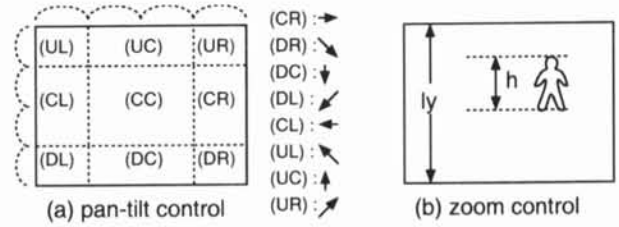

Figure 7: Camera control method

Since the camera moves briefly, and each zoom command changes magnification by only up to 15%, we decided to do without enlarging, reducing, and changing the template before and after a zooming.

3 Experiment

To test the proposed tracking algorithm, we monitored intruders from cameras installed on the top of a several story building.

We tried several scenes, confirming our method tracks a moving target securely. Requiring numerous pans, tilts, and zooms, the system switched its tracking mode accordingly.

Furthermore, the time between execution of a camera control command, and the end of the camera's vibration following the execution, was only 0.23 to 0.3 second approximately. Since the current processing rate is 2 frames per cycle, the system must track a moving object by template matching alone, without using frame differences for 4 to 5 cycles. This experiment confirmed the system is able to track a moving object, even during 4 to 5 such cycles.

4 Discussion

The proposed tracking algorithm uses the two methods complementarily. The template position is corrected by the frame difference data. The frame difference is calculated in the limited region around the template position. Furthermore, the system switches to the best tracking mode automatically. For this reason, the proposed system:

- Is able to track a target securely for long periods, even if the target image changes its shape, aspect, size, etc.;
- Can handle scenes where the target goes behind a static, concealing object, then reappears; and
- Is robust to changes in background texture, unless the target shows very poor contrast with the background.

Nevertheless, the algorithm has the following unsolved problems:

- The target's shadow affects the tracking significantly;
- Some motion of the target can destabilize the frame differences, causing erroneous corrections to the point to track, and wrong estimates of the target's size;
- The algorithm cannot tell whether the target is stationary, or concealed by another object; and
- It is not able to operate correctly when objects move close to the target or cross it.

If we can improve the algorithm to clearly define the target, including its movements, and thereby discount those template differences that do not comply with the corresponding model, then we should be able to solve such problems as caused by the target's shadow, and errors contained in the frame differences themselves. To distinguish whether the target is at a halt or wholly concealed, and to solve the problems caused by another object moving close to the target or crossing it, we need to establish procedures that correctly recognize such situations, using the templates accumulated so far.

5 Summary

We have proposed a tracking algorithm with camera control, for use in intruder monitoring systems. We conducted an outdoor tracking experiment using a sequence of images obtained from a pan-tilt-zoom camera installed on the top of a several story building. This experiment confirms the basic performance of the proposed system. We are now constructing long term estimation system to work night and day, to gather practical data over several months. We will improve the algorithm, based on these data.

References

[1] Shigeki Nagaya, Takafuji Miyatake, Takehiro Fujita, Wataru Ito, Hirotada Ueda, "Moving Object Detection by Time-Correlation-Based Background Judgement Method", *IEICE–D–II*, Vol. J79–D–II, No. 4, pp. 568–576, 1996. (in Japanese)

[2] E. Grimson, "A Forest of Sensors," Proc. of VSAM Workshop, November 1997.

[3] Chris Stauffer and W.E.L Grimson, "Adaptive background mixture models for real-time tracking", CVPR99, Fort Colins, CO, June 1999.

[4] W.E.L.Grimson, C. Staufffer, R. Romano and L. Lee, "Using adaptive tracking to classify and monitor activities in site," Proc. CVPR, 1998, pp. 22–29, 1998

[5] L.Davis, "Visual Surveillance and Monitoring," Proc. of VSAM Workshop, November 1997.

[6] H.Inoue, T.Tachikawa, M.Inaba, "Robot Vision System with a Correlation Chip fir Real-time Tracking, Optical Flow and Depth Map Generation," Proc. IEEE International Conferrence on Robotics and Automation, pp.1621-1626, 1992.

[7] Yasushi Mae, Shinya Yamamoto, Yoshiaki Shirai, and Jun Miura, "Optical Flow Based Realtime Object Tracking by Active Vision System," Proc. 2nd Japan-France Congress on Mechatronics, vol.2, pp.545-548, 1994.

[8] Shinya Yamamoto, Yasushi Mae, Yoshiaki Shirai, and Jun Miura, "Realtime Multiple Object Tracking Based on Optical Flows," Proc. 1995 IEEE Int. Conf. on Robotics and Automation, Vol.3, pp.2328-2333, 1995.

[9] Wataru Ito, Hirotada Ueda, "An autonomous object tracking camera with built-in image recognition hardware", *Proc. 5th Symposium on Sensing via Image Information*, Yokohama, Japan, pp. 13–17, June 1999. (in Japanese)

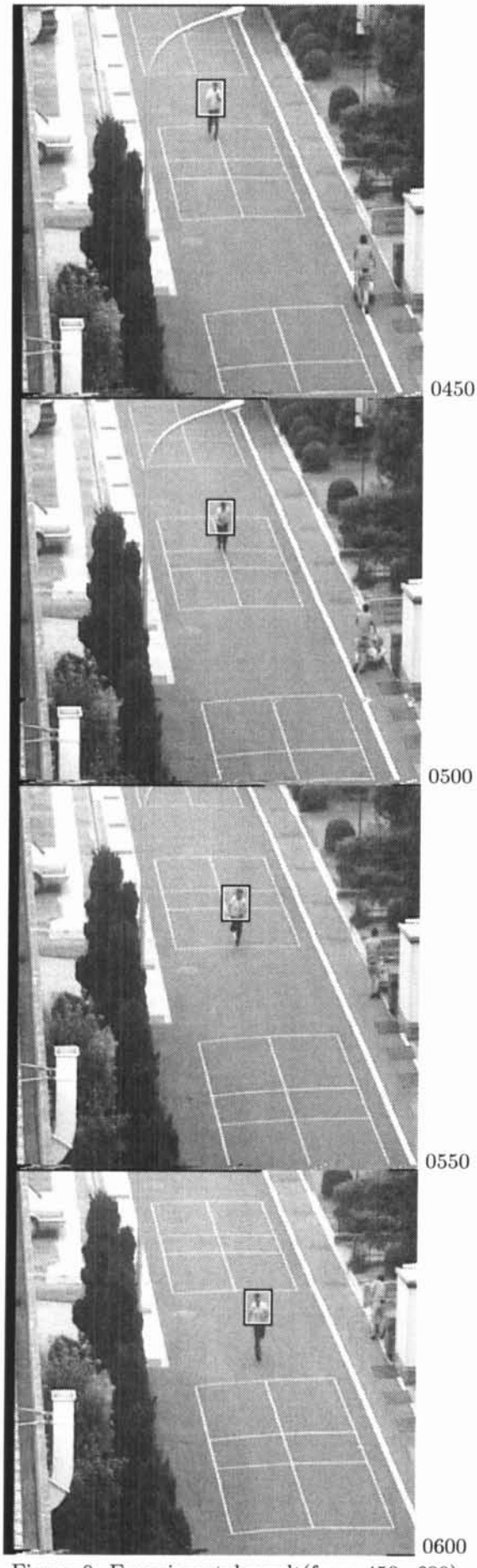

Figure 8: Experimental result(frame450~600)

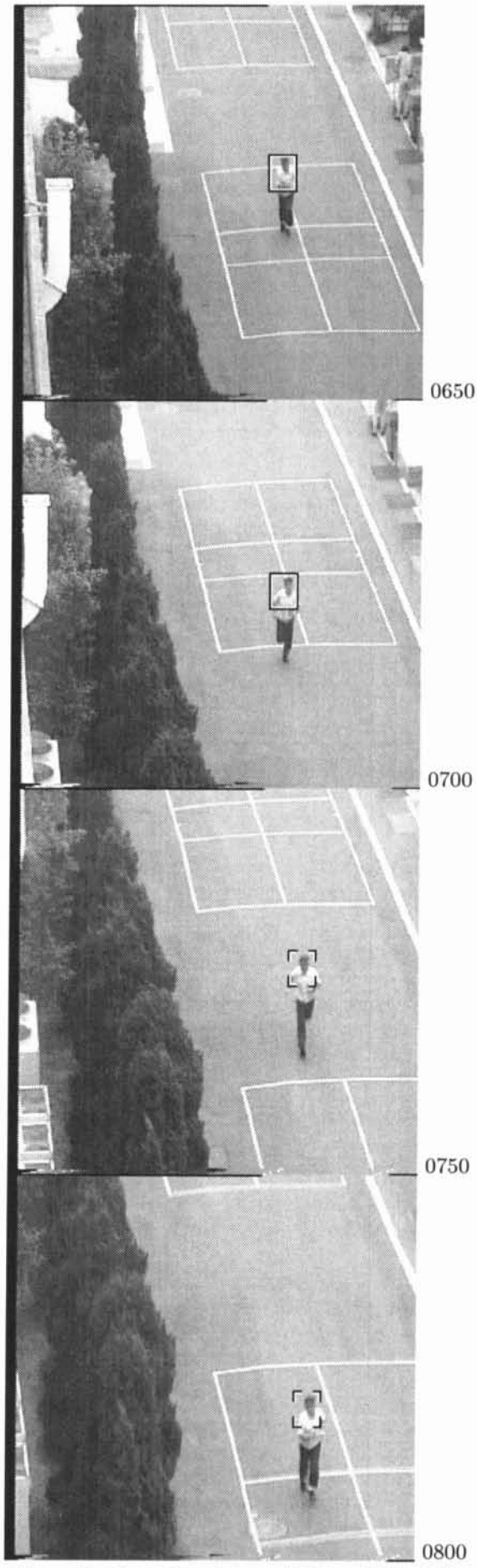

Figure 9: Experimental result(frame650~800)