MVA '96 IAPR Workshop on Machine Vision Applications, November. 12-14, 1996, Tokyo, Japan

Correlation-Based Visual Tracking enhanced by Affine Motion Description

Yoshihisa Adachi, Minoru Asada *
Department of Mech. Eng. for Computer-Controlled Machinery
Osaka University

Takayuki Nakamura †
Computer Science Department
Brown University

Abstract

Since conventional correlation-based visual tracking algorithms are based on a single reference block, they often fail to track a target in the following situations: 1) a view of the target image changes, and 2) a part of the target image is occluded. This paper proposes a correlation-based visual tracking method which copes with these problems based on affine motion description. A reference block to be tracked is covered by multiple windows, which are partly overlapped each other. The motion of the reference block is approximated by an affine motion description by applying the least mean square method to motion vectors each of which is estimated from each tracking window. Based on the estimated parameters of affine transformation, the method detects the change in the view of the target image and updates the reference block. It also finds when and where the occlusion occurs, and keeps the reference block. Some results for real image sequences are given to show the validity of the method.

1 Introduction

The robust visual tracking system would be very useful for various kinds of applications in robotics, human's gesture and expression recognition, and so on. As one of the visual tracking algorithms for realizing such a system, correlation techniques were developed [1]. In these techniques, a small area is initially captured from a image frame (this is called "reference block.") and the best matching block to this is searched for in subsequent image frames. To speed up the calculation of correlation, the search area is restricted around the neighbor of the current reference block. The representative techniques for correlation are based on the normalized cross correlation [2] and the sum of squared difference [3]. Most of these techniques do not deal with the following problems: 1) the view changes and 2) occlusions of the target image.

To cope with these problems, there have been some researches on correlation-based visual tracking. For the former, Darrell et al. [2] updated a reference block based on a normalized correlation value. However, their method does not solve the problem of occlusion. Nakamura et al. [4] utilized multiple tracking windows to solve the problem of the occlusion, but they do not cope with view changes of the target image.

In this paper, we propose a correlation-based visual tracking method enhanced by affine motion description. The motion of the reference block is approximated by an affine motion description by applying the least mean square method to motion vectors each of which is estimated from each tracking window. Based on the estimated parameters of affine transformation, the method detects the change in the view of the target image and updates the reference block. It also finds when and where the occlusion occurs, and keeps the reference block. Some results for real image sequences are given to show the validity of the method.

2 Visual Tracking enhanced by Affine Motion Description

2.1 Tracking process for each window

The tracking process consists of the following two procedures: to search the corresponding window which has the maximum correlation value evaluated by the following matching error (the Sum of Absolute Difference (SAD)), and to move the window to this location (see **Fig.**1).

$$Dist[i,j] = \sum_{k=0}^{K-1}\sum_{l=0}^{L-1} |R[k,l] - M[i+k, j+l]|,$$

$$i,j : 0 \leq i,j \leq 15,$$

where $R[x,y]$, $M[x,y]$, and $Dist[x,y]$ denote a reference window, a matching window, and an array of

*Address: 2-1, Yamadaoka, Suita, Osaka 565 Japan. E-mail: y-adachi@cv.ccm.eng.osaka-u.ac.jp

†Address: BOX 1910 115 Waterman St. Providence, RI 02906, USA. E-mail: tn@cs.brown.edu

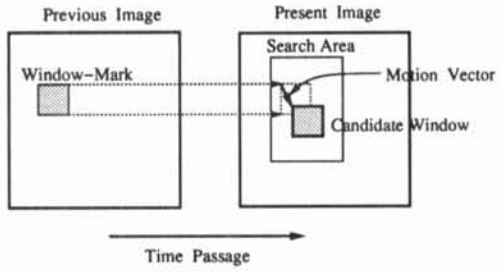

Figure 1: Tracking process for each window

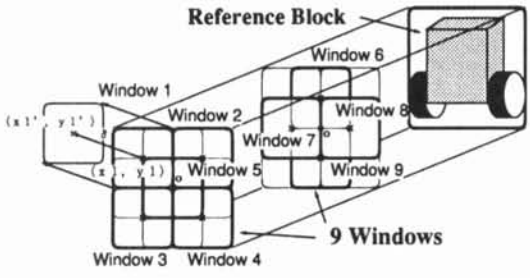

Figure 2: The arrangement of tracking windows

SAD, respectively. L and K denote the height and width of the reference window, respectively.

In our experiment, we can track about 100 windows (each window consists of 16 by 16 pixels) in real-time ($1/30sec$) by using a motion estimation processor (MEP) [5]. The search area is 32 by 32 pixels and the MEP outputs the location of each window where its SAD value is minimum.

2.2 Affine Motion Description

M(here $M = 9$) tracking windows are utilized to track one reference block, parts of which are overlapped each other (see **Fig.**2). The motion of the reference block is described by multiple motion vectors obtained from these tracking windows, and is approximated by an affine motion model.

Let $\boldsymbol{x}_k = [x_k \quad y_k]^T$ and $\boldsymbol{x'}_k = [x'_k \quad y'_k]^T$ be the center of each window in the first frame and that of each window in the second frame, respectively. Here, these image coordinates are measured with respect to the centroid of multiple tracking windows. Each motion vector $\boldsymbol{\delta}_k = [x'_k - x_k \quad y'_k - y_k]^T$.

Based on an affine motion model, the relationship between $\boldsymbol{x}_k$ and $\boldsymbol{x'}_k$ can be approximated as follows.

$$\boldsymbol{x'}_k = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{d}, \qquad (k = 1 \sim M)$$

where

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

is a deformation matrix, and $\boldsymbol{d}$ is the translation vector of the centroid of multiple windows.

Given $\boldsymbol{x}_k$ and $\boldsymbol{\delta}_k$, we can estimate $\boldsymbol{A}$ and $\boldsymbol{d}$ by the weighted least mean square method which minimizes

$$E = \sum_{k=1}^{M} \epsilon_k,$$

where

$$\epsilon_k = \{\boldsymbol{x'}_k - (\boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{d})\}^2. \qquad (1)$$

Using estimated $\boldsymbol{d}$, our method updates the location of the reference block between successive video frames.

2.3 Updating the Reference Block

Even if the target object does not move in the real world, the minimum SAD value fluctuates during a tracking process due to the video noise. Therefore, the outputs from tracking windows frequently change from -2 to $+2(pixels)$ caused by such fluctuation. Then we set up the following range for each parameter that tolerates the fluctuation:

$$\begin{aligned} 0.75 \leq\ & a_{11} \leq\ 1.25 \\ -0.25 \leq\ & a_{12} \leq\ 0.25 \\ -0.25 \leq\ & a_{21} \leq\ 0.25 \\ 0.75 \leq\ & a_{22} \leq\ 1.25 \end{aligned} \qquad (2)$$

Based on the range of each motion parameter, our method determine when the reference block should be updated. As long as the inequality (2) hold, no update, else update the reference block.

2.4 Coping with the view changes due to 3D motion

The use of an affine motion model is useful for approximating the two dimensional motion of the target, but it is not appropriate for the approximation of three dimensional motion. For an example, when a target turns round on a vertical axis in the real world, the reference block includes the image of background during the tracking process as shown in **Fig.**3. As a result, the visual tracking often fails.

Here, we deal with the problem of view changes of target which turns round on a vertical axis. In such a situation, $\boldsymbol{A}$ could be

$$\begin{bmatrix} a_{11} & 0 \\ 0 & 1 \end{bmatrix} \quad and \quad a_{11} < 1.$$

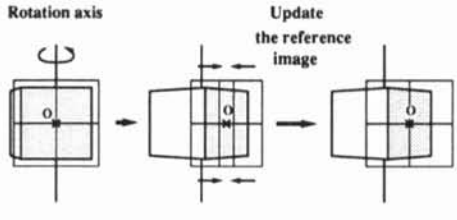

Figure 3: The tracking process during a target makes a turn on a vertical axis

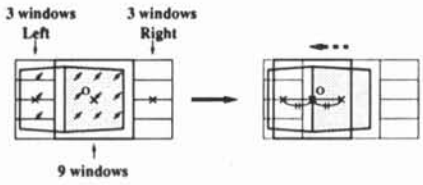

Figure 4: The arrangement of windows both sides

From this $\boldsymbol{A}$, we can detect the event of the rotation.

In order to detect the direction of the movement in the image plane, we arrange N tracking windows on both left and right side of the reference block when $a_{11} < 1$. **Fig.**4 shows an arrangement of new tracking windows. In our experiment, we set $N = 3$.

Then, during successive video frames, we calculate the squared of mahalanobis' distance $D_l^2(i)(i = 1 \sim 3)$ $(D_r^2(i))$ in terms of the motion vectors which are obtained from M tracking windows located at the center of the reference block and N windows on left side (or right side) of the reference block in order to evaluate the similarity of motion vectors between center and each side region.

$$D_l^2(i) = \begin{bmatrix} \delta_{xi} - \bar{\delta_x} & \delta_{yi} - \bar{\delta_y} \end{bmatrix} \begin{bmatrix} {s_x}^2 & s_{xy} \\ s_{xy} & {s_y}^2 \end{bmatrix}^{-1} \begin{bmatrix} \delta_{xi} - \bar{\delta_x} \\ \delta_{yi} - \bar{\delta_y} \end{bmatrix}$$

where δ_{xi}, δ_{yi}, $\bar{\delta_x}$, $\bar{\delta_y}$,s_x,s_y and s_{xy} are the movements of each side region $(i = 1 \sim 3)$, the averages of δ_x and δ_y, the variances of δ_x and δ_y and the covariance between δ_x and δ_y of center region, respectively.

If $\sum_{i=1}^{3} D_l^2(i) < \sum_{i=1}^{3} D_r^2(i)$,
then the search area moves to left (see **Fig.**5(a)),
else the search area moves to right (see **Fig.**5(b)).

2.5 Coping with Occlusions

When some of M tracking windows are occluded or affected by noise, those motion vectors do not correctly reflect the motion of the target object.

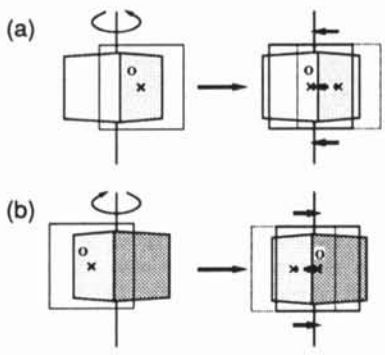

Figure 5: The movement of the search area

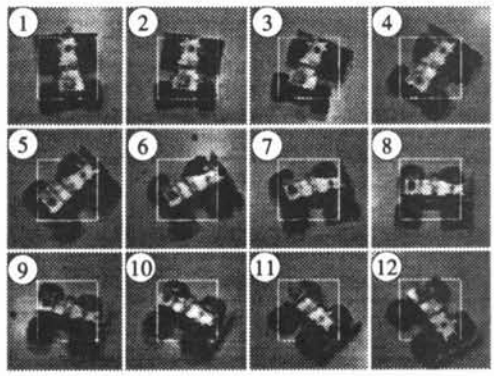

Figure 6: The acquired reference blocks

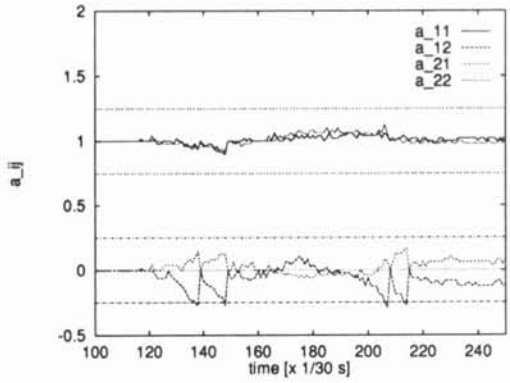

Figure 7: The each component value of the A

Therefore, the occluded windows cause much damage to the estimated $\boldsymbol{A}$. If such occluded windows could be discriminated, we can estimate $\boldsymbol{A}$ correctly.

In order to find out which windows are occluded windows, we evaluate the residual ϵ_k that are obtained from eq.(1) because the residuals of occluded windows are larger than that of not occluded windows.

Using cluster analysis (K-mean method) [6] based on the residuals of all windows, we divided all windows into two groups. One group with smaller residual value corresponds to the non-occluded windows and the other with larger value corresponds to the occluded ones. Then, using the weighted least mean square method, we estimate $\boldsymbol{A}$ and $\boldsymbol{d}$ again setting weights of the method as follows:

If a window is regarded as the occluded one,
then set the weights with small values,
else set the weights with large values.

Owing to evaluating the residuals, the tracking performance is not much affected by the initial pattern of reference block. Even if some of windows are occluded or affected by noise, the method finds out which windows are occluded or affected and can continue to track the target based on the information

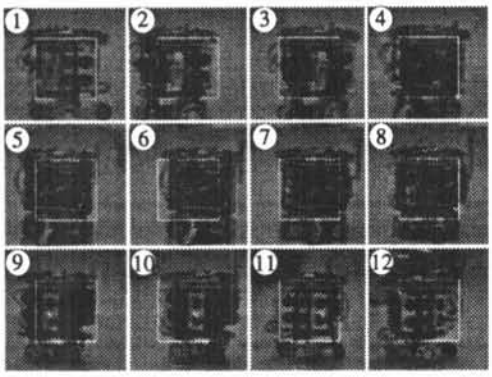

Figure 8: The acquired reference blocks

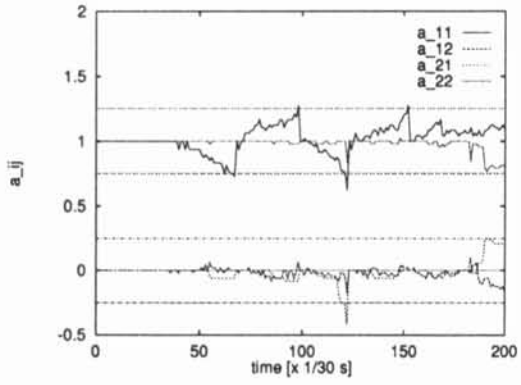

Figure 9: The each component value of the A

obtained from not occluded windows.

3 Experimental results

In case of view changes

Fig.6 and 7 show the reference blocks automatically acquired from the image sequence and the changes in each component value of $\boldsymbol{A}$ during the tracking process, respectively. In this image sequence, a mobile robot moves on the ground and its view changes frequently. In **Fig.**7, the component a_{12} exceeds the range in (2) at time steps 138, 148, 207 and 214, just after these steps, the reference block is updated as image numbers 2, 3, 4 and 5 in **Fig.**6, respectively.

Fig.8 and 9 show the reference blocks automatically acquired from the image sequence and the changes in each component value of $\boldsymbol{A}$ during the tracking process, respectively. In this image sequence, the mobile robot turns round on the vertical axis in the real world. In **Fig.**9, the component a_{11} exceeds the range in (2) at time step 67, just after this step the reference block is updated as 2 in **Fig.**8.

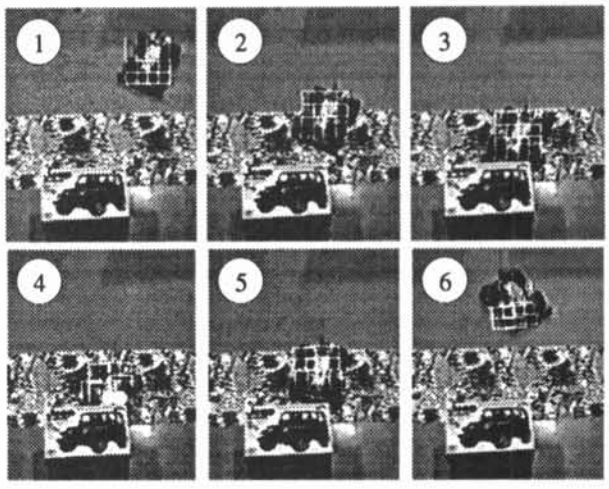

Figure 10: The results for coping with the occlusion

In case of occlusions

Fig.10 shows a sequence of images in which the target is tracked by our method coping with occlusions. In this figure, white circles show the windows detected as occluded ones.

4 Conclusions

In **Section.**2.4 we only deal with the problem of view changes of target which turns round on vertical axis. Generally, we can cope with the rotation of which axis is parallel to the image plane by applying the principal component analysis. We expect the system to cope with other cases by regarding them as view changes.

References

[1] Jerome Martin and James L. Crowley "Comparison of Correlation Techniques". In *Intelligent Autonomous Systems*, pages 86–93, 1995.

[2] Trevor J. Darrell and Alex P. Plentland "Recognition of Space-Time Gestures using a Distributed Representation". In *M.I.T. Media Laboratory Vision and Modeling Group Technical Report No.197*

[3] . N. Papanikolopouslos, P. K. Khosla, and T. Kanade "Visual tracking of a moving target by a camera mounted on a robot: a combination of control and vision". In *IEEE Transactions on Robotics and Automation*, vol. 9, pages 14–32, 1993.

[4] T. Nakamura and M. Asada "Motion Sketch: Acquisition of Visual Motion Guided Behaviors ". In *IJCAI'95 Vol.1*, pages 126–132, 1995.

[5] H. Inoue, T. Tachikawa, and M. Inaba. "Robot vision system with a correlation chip for real-time tracking, optical flow and depth map generation". In *Proc. IEEE Int'l Conf. on Robotics and Automation*, pages 1621–1626, 1992.

[6] . R. Duda, and P. Hart "Pattern Classification and Scene Analysis". *John Wiley and Sons, Inc.*, 1973.