Image Processed Tracking System of Multiple Moving Objects Based on Kalman Filter

Dong-Kyu Kim and Sang-Bong Kim*
Department of Mechanical Engineering, Pukyong National University
San 100, Yongdang-dong, Nam-gu, Pusan 608-739, Korea
Hak-Kyeong Kim
Department of Mechatronics Engineering, Pukyong National University
San 100, Yongdang-dong, Nam-gu, Pusan 608-739, Korea

This paper presents a development result for image processed tracking system of multiple moving objects based on Kalman filter and a simple window tracking method. The proposed algorithm of foreground detection and background adaptation (FDBA) is composed of three modules: a block checking module (BCM), an object movement prediction module (OMPM), and an adaptive background estimation module (ABEM). The BCM is processed for checking the existence of objects. To speed up the image processing time and to precisely track multiple objects under the object's mergence, a concept of a simple window tracking method is adopted in the OMPM. The ABEM separates the foreground from the background in the reset simple tracking window in the OMPM. It is shown through experimental results that the proposed FDBA algorithm is robustly adaptable to the background variation in a short processing time. Furthermore, it is shown that the proposed method can solve the problems of mergence, cross and split that are brought up in the case of tracking multiple moving objects.

Key Words: Kalman Filter, Foreground Detection, Background Adaptation, Tracking, Multiple Moving Objects

1. Introduction

In the most image sequence processing systems for tracking target objects, the first important step is to separate the foreground from the background and then to detect precisely the motions regardless of their speed, direction or texture. The second step is to minimize the tracking error and to get the stable tracking (Karmann et al., 1990 and Nagel, 1987). There are several methods to track object and to extract its trajectory: a method based on the optical flow (Nagel, 1987), a simple window tracking method (Jee, 1994), difference image method, maximum greyvalue holding method (Kim, 1998 and Wu, 1996), Kalman filter method (Park, 1998; Gonzalez, 1992) and so on. But these approaches did not consider handling objects under the variation of the background. Karmann and Brandt, 1990, presented an algorithm in which the process of the greyvalue variation in a background image sequence is described as a signal processing system. Disadvantages of the Karmann and Brandt's approach are that the calculation time of the algorithm is long due to the process of the full-size image and the implementation of the algorithm is very complex in actual application because of many assumptions. Furthermore, only separating foreground from background in a single object is done and the tracking is never considered.

In this paper, in order to realize short pro-

* Corresponding Author,
E-mail: giberdon@dreamwiz.com
TEL: +82-51-620-1606; **FAX**: +82-51-621-1411
Department of Mechanical Engineering, Pukyong National University, San 100, Yongdang-dong, Nam-gu, Pusan 608-739, Korea. (Manuscript **Received** July 29, 2000; **Revised** November 14, 2001)

cessing time and to robustly track the multiple target objects under background noise or arbitrary disturbance, an image processed tracking algorithm incorporating the Kalman filter with a simple window tracking method is presented. The Kalman filter is used for three objectives such as background adaptation under noises or disturbance, object detection in block checking procedure and object tracking to reset the simple window according to the movement of the detected target object. The simple window tracking method is adopted to speed up the tracking time and to precisely track multiple objects under mergence, cross and split. The developed algorithm called FDBA (foreground detection and background adaptation) algorithm includes three modules: a block checking module (BCM), an object movement prediction module (OMPM), and an adaptive background estimation module (ABEM). The BCM based on the Kalman filter is processed for detecting the appearances of new objects in sight of CCD camera. Recognized objects are modeled under assumptions that the moving objects have a constant distance in z direction and move with constant velocity in x and y directions. This model is used for Kalman filter design in order to predict the position of objects in the OMPM. ABEM separates the foreground from the background inside the simple window. The background dynamics of each pixel is modeled and its estimation is realized by the Kalman filter.

It is shown through some experimental results that the proposed FDBA algorithm is robustly adaptable to the variation of the background affected by arbitrary disturbance and it takes short processing time in spite of tracking multiple objects.

2. Modeling of Multi-Dimensional Moving Objects

This section presents a dynamic model for describing the motion of multi-dimensional moving objects. The dynamic equation for multi-dimensional moving objects can be represented as follows;

$$m_i \frac{d^2\psi_i}{dt^2} + \mu_i \frac{d\psi_i}{dt} = f_i(i) \tag{1}$$

where, $\psi_i = [x_i\ y_i\ z_i]^T$ denotes coordinate axes in x, y and z direction for each object ($i=1, 2, \cdots, n$), m_i is its mass, μ_i is its damping coefficient and $f_i(t)$ is its external force at time t.

In image processing system for a moving object captured by CCD camera, the damping and external forces in Eq. (1) can be ignored, then Eq. (1) can be rewritten by

$$m_i \frac{d^2\psi_i(t)}{dt^2} = 0 \tag{2}$$

We assume that a moving object keeps the constant distance in the z direction and moves with constant velocities in the x and y directions. If the sampling time, Δt, is sufficiently small, the first derivative of ψ_i can be rewritten as follows;

$$\frac{d\psi_i(t)}{dt} \cong \frac{\psi_i(t+\Delta t) - \psi_i(t)}{\Delta t} \tag{3}$$

Now, let us define

$$\dot{\psi}_i(k) \triangleq \frac{d\psi_i(t)}{dt}\Big|_{t=k\Delta t}, \quad (k=0, 1, 2, \cdots) \tag{4}$$

Using $t=k\Delta t$ and substituting (3) into (4), Eq. (4) is rewritten by

$$\dot{\psi}_i(k) = \frac{\psi_i[(k+1)\Delta t] - \psi_i(k\Delta t)}{\Delta t} \tag{5}$$

For simplicity, let $(k+1)\Delta t$ be $(k+1)$ in Eq. (5). Then from Eq. (5), the following discretized equation can be obtained.

$$\psi_i(k+1) = \dot{\psi}_i(k)\Delta t + \psi_i(k) \tag{6}$$

Let us define the state variable vector as

$$\zeta_i(t) \triangleq [x_i(t)\ \dot{x}_i(t)\ y_i(t)\ \dot{y}_i(t)]^T \tag{7}$$

Discretizing the state vector ζ_i with sampling time, Δt using Taylor's series expansion and difference approximation method, the system equation is given by

$$\zeta_i(k+1) = A\zeta_i(k) \tag{8}$$

where, $A = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix}$

The output equation is given by

$$\gamma_i(k) = C\zeta_i(k) \tag{9}$$

where, $C=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$

We can see that the system equation for each object does not relate to its mass. Therefore, all the system equations for multiple objects have the same form. The system output, $\gamma_i(k)$ is given by the positions in x and y directions which are calculated at each sampling time k by image processing measurement using a stationary CCD camera with fixed focal length.

3. FDBA (Foreground Detection and Background Adaptation) Algorithm

The FDBA algorithm is to track the moving object in image sequences with the variations of the background and the illumination based on the Kalman filter. It also can track multiple moving objects and can overcome some problems such as mergence, cross and split generated in case of tracking multiple moving objects. That is, when a dynamic model is used by Kalman filter in order to predict the positions of objects, the next-step positions of objects become known. So objects are separated properly by comparing predicted position with measured position after one step process when objects are merged, split and crossed with each other. This algorithm is composed of BCM (Block Checking Module), OMPM (Object Movement Prediction Module) and ABEM (Adaptive Background Estimation Module). The following subsections will explain the detailed functions of each module and step-by-step algorithm of the FDBA.

3.1 Object tracking algorithm using kalman filter

The system dynamic equations of the previous Sec. 2, which are linearized in sufficiently small time intervals, are rewritten here by including noises as

$$\zeta_i(k+1) = A\zeta_i(k) + w(k) \tag{10}$$
$$\gamma_i(k) = C\zeta_i(k) + v(k) \tag{11}$$

where, $\zeta_i(k)$ is the state variable vector, $v(k)$ and $w(k)$ are the uncorrelated white noise vectors that are characterized by zero-mean and by the covariance matrices Q and R as follows;

$$E[w(\mathrm{k})]=0,\ E[v(k)]=0 \tag{12}$$
$$E[w(k)w(j)^T]=Q\delta(k, j) \tag{13}$$
$$E[v(k)v(j)^T]=R\delta(k, j) \tag{14}$$
$$E[w(k)v(j)^T]=0 \tag{15}$$

where, $E[\cdot]$ denotes the mean value and $\delta(k, j)$ denotes the Kronecker delta function.

Usually, the Kalman filtering method is realized by assumption that the best information of a system state is obtained by an estimation, which explicitly takes account of the measurement noise of the system state values.

The estimation at time k is given by

$$\hat{\xi}_i(k) = \tilde{\xi}_i(k) + K_i(k)[\gamma_i(k) - C\tilde{\xi}_i(k)] \tag{16}$$

with a prediction term

$$\tilde{\xi}_i(k) = A\hat{\xi}_i(k-1) \tag{17}$$

where $\gamma_i(k)$ is the system output and becomes the input for Kalman filter. $K_i(k)$ is the Kalman gain matrix.

For the computer application, the discrete Kalman filter must be implemented. The discrete Kalman filter has two steps at each time k. One is the *time update*, by which $\hat{\xi}_i(k-1)$ is updated to $\hat{\xi}_i(k/k-1)$, and the other is *measurement update*, by which the system output $\gamma_i(k)$ at time k is incorporated to provide the updated estimate $\hat{\xi}_i(k)$. The Kalman filter equations for the discrete system are represented as follows;

Time update: (Effect of system dynamics)

Error covariance:

$$P_i(k+1/\mathrm{k}) = AP_i(k)A^T + Q \tag{18}$$

Estimate update:

$$\hat{\xi}_i(k+1/k) = A\hat{\xi}_i(k) \tag{19}$$

Measurement update:
(Effect of measurements $\gamma_i(k)$)

Error covariance:

$$P_i(k+1) = (I - K_i(k+1)C)P_i(k+1/k) \tag{20}$$
$$K_i(k+1) = P_i(k+1)C^T R^{-1} \tag{21}$$

Estimate update:

$$\hat{\xi}_i(k+1) = \hat{\xi}_i(k+1/k)$$
$$+K_i(k+1)(\gamma_i(k+1)$$

$$-C\hat{\xi}_i(k+1/k)) \tag{22}$$

The computations of the gain matrix, involving both the estimate recursion and the error covariance recursion in Eqs. (18) ~ (22), may be performed on-line in real time.

3.2 Kalman filter for background adaptation

The process of greyvalue variation in a background image sequence is described as a signal processing system (for each pixel). The system state is represented by the greyvalue and the first derivative of the greyvalue. The background dynamics of each pixel in this system is modeled as follows;

$$\xi_i(k+1) = A_b\xi_i(k) + w_b(k) \tag{23}$$

$$\eta_i(k) = C_b\xi_i(k) + v_b(k) \tag{24}$$

where,

$$\xi_i(k) = \begin{bmatrix} s(k) \\ \dot{s}(k) \end{bmatrix},\ A_b = \begin{bmatrix} 1 & 0.7 \\ 0 & 0.7 \end{bmatrix},\ C_b = [1 \quad 0]$$

$\xi_i(k)$ is the state variable vector, A_b is the state transition matrix given by Karmann et al, 1990, C_b is the measurement matrix, $\eta_i(k)$ is the output variable vector, $v_b(k)$ and $wb(k)$ are denoted as the uncorrelated white noise vectors that are characterized by zero-mean and by the covariance matrices Q_b and R_b.

The background estimation at time k is given by the following equation similar to Eq. (16);

$$\hat{\xi}_i(k) = \bar{\xi}_i(k) + K_b(k)[s_i(k) - C_b\bar{\xi}_i(k)] \tag{25}$$

where,

$$\hat{\xi}_i(k) = \begin{bmatrix} \hat{s}(k) \\ \dot{\hat{s}}(k) \end{bmatrix}$$

with a prediction term

$$\bar{\xi}_i(k) = A_b\hat{\xi}_i(k-1) \tag{26}$$

The greyvalue of a pixel, $s(k)$ of the frame at sampling time k given by a stationary CCD camera with fixed focal length is put into the system input for Kalman filter. $\dot{s}(k)$ is the greyvalue variety. The system state at time k is represented by hat $\hat{s}(k)$ corresponding to the estimated pixel background greyvalue in position (x, y), and $\dot{\hat{s}}(k)$ is the estimated variety. The system is observable although it is unstable. So the Kalman filter can be applied to this system sufficiently.

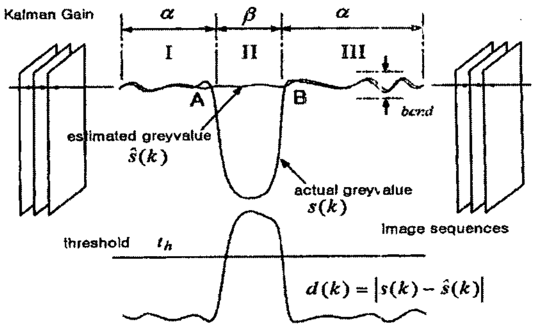


Fig. 1 Concept of Kalman filtering to image sequence

To implement the Kalman filter for the background adaptation, the Riccati equation may be solved in off-line before the implementation to compute the error covariances and the Kalman gain, $K_b(k)$. The Kalman gain may then be stored in computer memory. The estimating recursion in equations (20) and (21) is performed by using the stored values of $K_b(k)$.

In this study, as shown in Fig. 1, two Kalman gain values α and β are obtained in off-line for clearly separating the foreground from the background. That is, at point A, if the variation of the actual greyvalue, $s(k)$, is larger than the *band* in Fig. 1, the Kalman gain for slow estimation of the actual greyvalue is chosen as β. Otherwise, the Kalman gain for fast estimation of the actual greyvalue is α. When the gain β is adopted to the Kalman filter in the region II, the absolute value of the difference, $d(k)$ between the actual greyvalue, $s(k)$ and the estimated greyvalue, hat $\hat{s}(k)$ becomes large. So the foreground can be separated from the background by comparing the difference, $d(k)$ with the threshold, t_h. The Kalman gain for the background adaptation is given then by the following image processing condition;

$$K_b(k) = \begin{cases} \alpha \text{ when Trigger } (k) = 0 \\ \beta \text{ when Trigger } (k) = 1 \end{cases} \tag{27}$$

$$\text{Trigger } (k) = \begin{cases} 0 & \text{if } \{|s(k)-s(k-1)| > band \text{ AND } sign(s(k)-s(k-1)) > 0\} \text{ OR } m(k-1) = 1 \\ 1 & \text{if } \{|s(k)-s(k-1)| > band \text{ AND } sign(s(k)-s(k-1)) < 0\} \text{ OR } m(k-1) = 0 \end{cases}$$

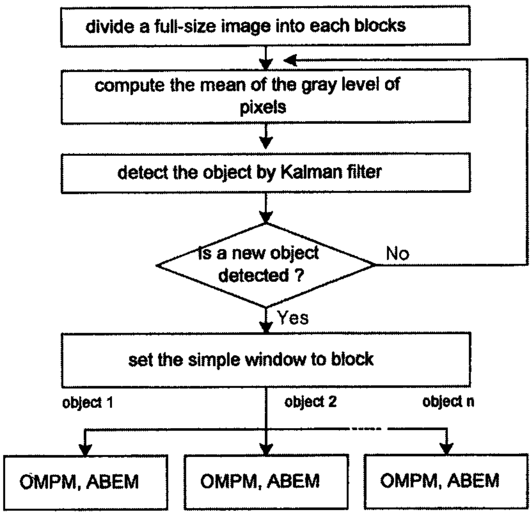


Fig. 2 Flowchart of BCM

$$m(k-1)=\begin{cases}1 & \text{if } d(k-1)\geq t_h \\ 0 & else\end{cases} \tag{28}$$

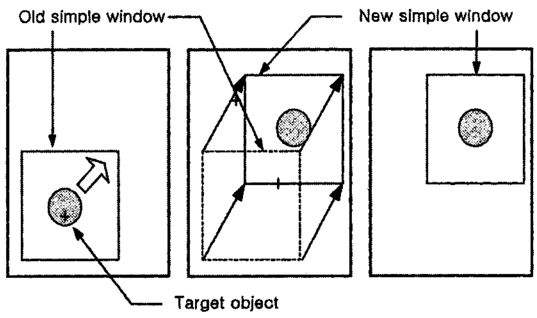
$$d(k-1)=|s(k-1)-\hat{s}(k-1)| \tag{29}$$

$$t_h=const,\ \forall x,\ y,\ k \tag{30}$$

where, Trigger(k) is introduced to divide the region I, II and III in Fig. 1. $m(k-1)$ is a binary image representing the segmentation of the pixel at position (x, y) at time $k-1$. The greyvalue is 1 for the foreground and 0 for the background. And t_h denotes a threshold value.

3.3 Block checking module

Figure 2 shows the flowchart of the BCM. In order for the BCM to check existence of multiple moving objects, the full-size image is divided into the appropriate blocks depending on the size of the target objects, and the mean greyvalue of the pixels inside those blocks is taken. When a suddenly changed greyvalue in a block is checked by the Kalman filter, we can judge that there exists an object to be tracked. If a new object is searched in any blocks, the simple window is set to the blocks for the OMPM and the ABEM. The BCM estimates the mean greyvalue in each block which is regarded as a large sized pixel. So the matrix operations implemented by the Kalman filter are less in the BCM than in the processing of the full-size image.



(a) (b) (c)

Fig. 3 Concept of the simple window ; (a) At $k-1$ (b) Processing at k (c) Processed at k

3.4 Object movement prediction module

In this module, a new concept of a simple window is proposed for high speed image processing and real time tracking. Figure 3 shows the concept of the simple window for the object tracking. If the size of a target object is known, we can set a proper size window, called a simple window, around the target object. The target object in the old simple window moves from the position at time $k-1$ to the point at time k as shown in Fig. 3(a). At time k, the new simple window is reset around the predicted position with appropriate speed according to the given sampling time by using the Kalman filter explained in subsection 3.1 as shown in I of Fig. 3(b). The position of the target object in the new simple window is detected by ABEM as shown in Fig. 3(c). This module resets iteratively the new simple window for the ABEM with the information for the estimated position of the target object.

For multiple objects, the simple windows are set as many as the number of objects and each simple window is moved individually with each position and velocity. Furthermore, it is shown that the estimation for next-step positions of objects can be possible to solve the problems of mergence, cross and split that are brought up in case of tracking multiple moving objects.

3.5 Adaptive background estimation module

This module separates the foreground from the background in the reset simple window based on the algorithm which can be applied robustly for the background under the illumination change.

Figure 4 shows an example for the ABEM. When a car moves into the simple window, the foreground is separated from the background with the illumination variation by the daylight. The background dynamics of each pixel in the region inside the simple window is modeled as shown in subsection 3.2 and the background estimation is realized by Kalman filter.

When the Kalman filter is implemented in the computer application, the concept of the simple window plays an important role in reducing the matrix operations for each pixel inside regions including the moving objects. Figure 5 shows a flowchart of the OMPM and the ABEM. Figure 6 shows the concept of FDBA algorithm.

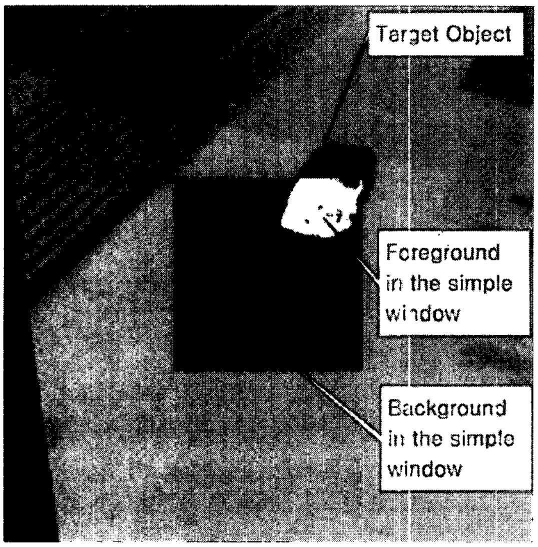


Fig. 4 Example of the ABEM

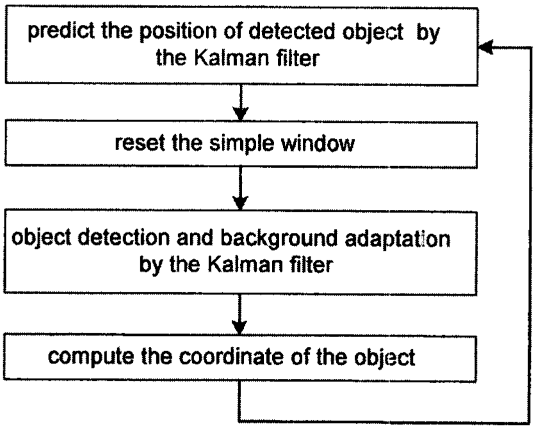


Fig. 5 Flowchart of OMPM and ABEM

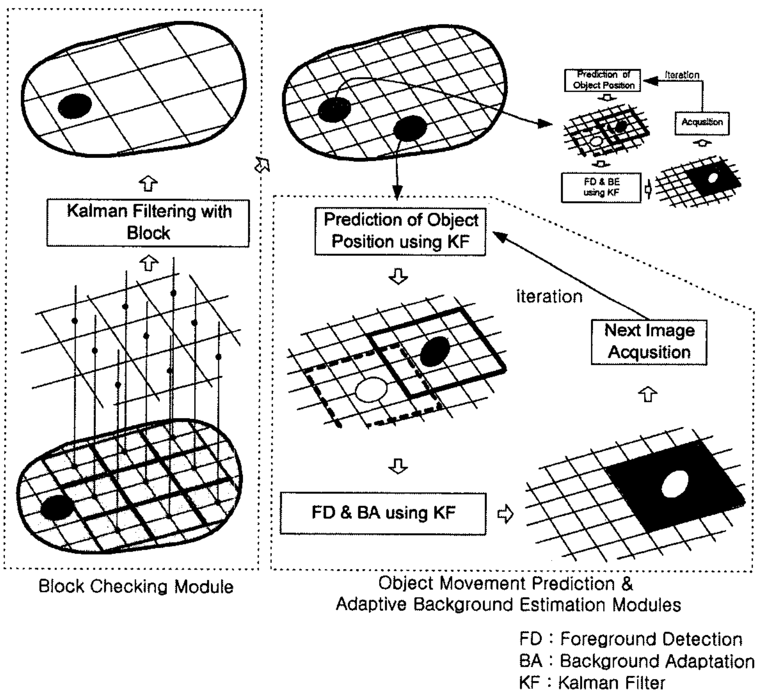
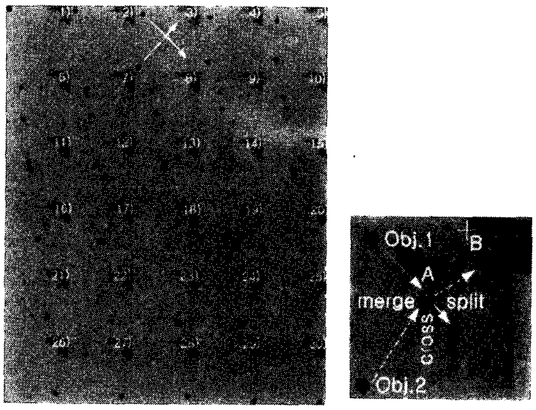


Fig. 6 Concept of FDBA algorithm



(a) (b)

(a) Sampled image sequences(30 frames)
(b) Crossing state

Fig. 7 Sampled image sequences and overlayed image(frame No. 3, 12 and 19)

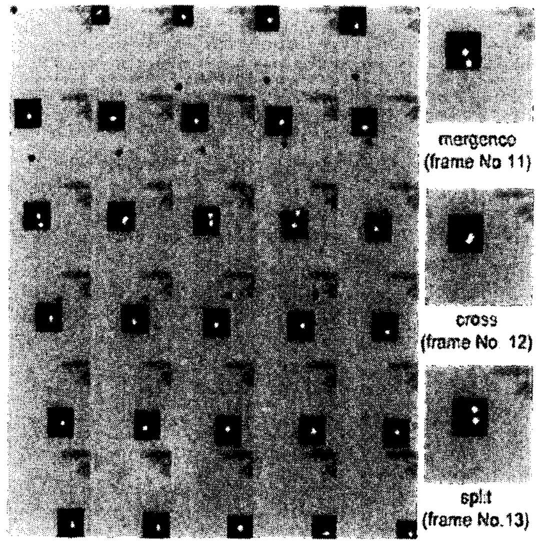


Fig. 8 Detected object in image sequences (Obj. 1)

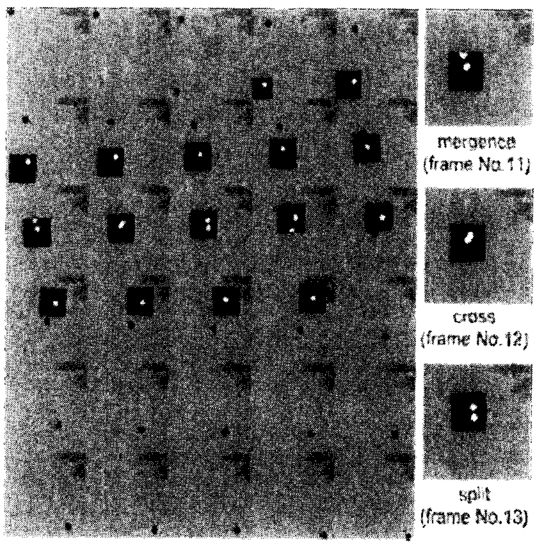


Fig. 9 Detected object in image sequences (Obj. 2)

4. Results and Consideration

The input image sequence is an image drawing arbitrary target object to be detected on the actual image in 100×100 (pixels) size. Figure 7 shows sampled image sequences and an overlayed image with frame No. 3, 12 and 19. The input image sequences of the 30 frames(6×5 frames) with 100×100 (pixels) size are shown in Fig. 7(a) and the numbers are frames numbers. Figure 7(b) shows the states of mergence, cross and spit as an overlayed image with frame No. 3, 12 and 19 in Fig. 7(a). Object 1 moves into lower right direction, and the other object 2 moves into upper right direction.

The FDBA is applied to the image sequences with the background variation and crossing of two objects without a collision. Figures 10 and 11 show that two objects can be tracked properly by the simple window(15 times 15 pixels) under the mergence, cross and split. Object 2 was checked in frame 3 by the BCM, and the ABEM and the OMPM were carried out iteratively for the background adaptation and the movement prediction respectively.

Figures 10 and 11 show the system behaviour and the estimation error, $d(k)$ in points A and B in Fig. 7. The target object passed through A point and leaves of a tree are moved by the wind to B point. In Fig. 10, the estimated background greyvalue at point A, dotted line is successfully estimated with the greyvalue of about 0.4 although the greyvalue of the actual object at point A, the continuous line is decreased to about zero. Figure 11 shows the absolute value of difference, $d(k)$ between actual greyvalue and estimated greyvalue of each frame in points A and B. The foreground can be separated from the background by comparing $d(k)$ with the threshold, t_h. So the object inside the simple window can be effectively detected under the background variation that is regarded as system noise (movement of the leaves and daylight).

Table 1 shows the processing time of the FDBA per one frame(100×100 pixels). From the results, the processing time of the proposed FDBA using one simple window(15×15 pixels) is about ten times faster than the processing time of the full-size image(100×100 pixels). In addition, the tracking of multiple moving object using two simple windows is about four times faster than the processing of the full image.

Table 1 Processing time per one frame

Tracking methods	Processing time
Karmann et al (the full-size image)	2.97sec/frame
FDBA (one simple window)	0.33sec/frame
FDBA (two simple window)	0.67sec/frame

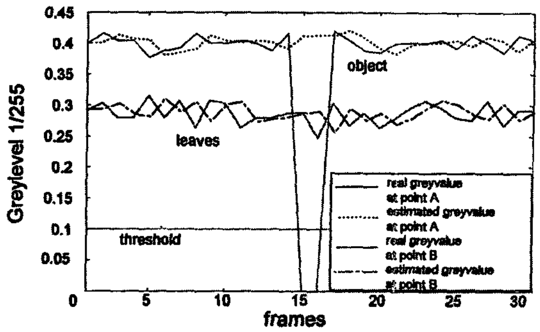


Fig. 10 Estimated values in points A and B

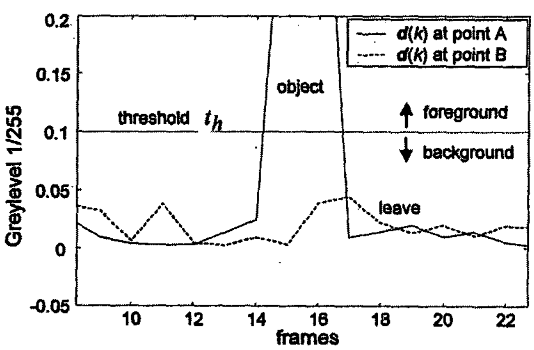


Fig. 11 $d(k)$ in points A and B

5. Conclusions

In this paper, in order to realize short processing time and to track robustly the multiple target objects under background variation, an image processed tracking algorithm, FDBA incorporating the Kalman filter wih a simple window tracking method is presented. The Kalman filter is used for three objectives such as the object detection, the background adaptation and the object tracking. The experimental results show that the proposed FDBA algorithm can be applied effectively for tracking multiple objects that are subject to the variations of the background and illumination, and can be processed with fast processing time. Furthermore, the proposed method can solve the problems of mergence, split and crossing brought up in the case of tracking multiple moving objects by predicting the next-step positions of objects.

References

Brofferio S. et al, 1990, "A Background Updating Algorithm for Moving Object Scenes," *Time-varying Image Processing and Moving Object Recognition*, Vol. 2, Elsevier Publishers B. V., Amsterdam, The Netherlands, pp. 289~296.

Chapman and Hall, *Image Precessing, Analysis and Machine Vision*, 2-6 Boundary Row, London SE1 8HN.

Gonzalez R. C., 1992, *Digital Image Processing*, Addison Wesley.

Gregory A. B., 1994, *Digital Image Processing*, Wiley and Sons.

Jee M. S. and Kim S. B., 1994, "Development of Two Dimensional Position Measuring for Floating Structure Using an Image Processing Method," *Journal of Ocean engineering and technology*, Vol. 8(2), pp. 22~27.

Karmann K. P., Brandt A. V., 1990, "Moving Object Recognition Using an Adaptive Background Memory," duddl*T*ime-varying Image Processing and Moving Object Recognition, Vol. 2, Elsevier Publishers B. V., Amsterdam, The Netherlands, pp. 297~307.

Kim, D. K., Kang, B. Ch. and Kim, S. B., 1998

"Development of Counting and Trajectory Extraction Algorithms of Sperm Using Image Processing," *13th KACC*, pp. 881~884.

Leung, M. K., and Yang, Y. H., 1987, "Human Body Motion Segmentation In A Complex Scene," *Pattern Recognition*, Vol. 20, pp. 55~64.

Lewis, F. L., *Applied Optimal Control & Estimation*, Prentice-Hall Interna-tional, 1994.

Nagel, H. H., 1987, "On the Estimation of Optical Flow: Relations Between Different Approaches and Some New Results," *Artificial Intelligence*, Vol. 33, pp. 299~324.

Park, H. S., Kim, H. S. and Kim S. B., 1998, "The Tracking of 2-D Moving Object Using the Digital Image Processing," *Proceedings of the KSPSE Autumn Annual Meetings '98*, pp. 175~180.

Samuel, B. and Robert, P., 1999, *Design and Analysis of Modern Tracking Systems*, Artech House

Tompson, W. B. and Barnard, S. T., 1981, "Low-Level Estimation and Interpolation Visual motion," *IEEE computer*, Vol. 14(8), pp. 20~28.

Wu, W. et al., 1996, "Analysis of Sperm Motion by Trajectory Extraction from Image Sequence," *Journal of the Society of Instrument and Control Engineers*, Vol. 32, pp. 597~603.

Wang, H. and Daley, S., 1996, "Actuator Fault Diagnosis: An Adaptive Observer Based Technique," *IEEE Trans. on Automatic Control*, Vol. 41, No. 7, pp. 1073~1078.