Efficient Tracking of a Moving Object using Optimal Representative Blocks

Wan-Cheol Kim, Cheol-Ho Hwang, and Jang-Myung Lee

Abstract: This paper focuses on the implementation of an efficient tracking method of a moving object using optimal representative blocks by way of a pan-tilt camera. The key idea is derived from the fact that when the image size of a moving object is shrunk in an image frame according to the distance between the mobile robot camera and the object in motion, the tracking performance of a moving object can be improved by reducing the size of representative blocks according to the object image size. Motion estimations using Edge Detection (ED) and Block-Matching Algorithm (BMA) are regularly employed to track objects by vision sensors. However, these methods often neglect the real-time vision data since these schemes suffer from heavy computational load. In this paper, a representative block able to significantly reduce the amount of data to be computed, is defined and optimized by changing the size of representative blocks according to the size of the object in the image frame in order to improve tracking performance. The proposed algorithm is verified experimentally by using a two degree-of-freedom active camera mounted on a mobile robot.

Keywords: Optimal representative blocks, perspective transformation, active camera.

1. INTRODUCTION

Many measurement devices have been created that are similar to the human senses. Some of them even demonstrate superior abilities than the human senses when applied to specific industries. Visual sensors have been recently developed due to improvements in computer performance necessary for the computation of enormous quantities of information. In fact, image tracking research has been applied to industrial robot vision, military precision guided weapons, ITS (Intelligent Traffic Systems) and IVS (Intelligent Vehicle Systems).

In this paper, we suggest a new tracking method for moving objects using mobile robots. Generally, for real-time object tracking, ED (Edge Detection) and BMA (Block Matching Algorithm) have traditionally been employed [1]. In the ED method, a great deal of time is spent on object recognition because of the numerous pixels in the high-resolution images. Specifically, in a convolution operation step, the system operations related to the memory take too much time and then there are several cases where real-time vision data are lost because of slow image processing. In the BMA method, motion estimation is calculated with the matched block information [2-4]. This includes motion estimation by investigating the maximum correlation between a former frame (n-1) block data and a present frame (n) block data. MAD (Mean Absolute Difference) and MSD (Mean Squared Difference) are generally utilized for the BMA [5-8]. The BMA method has a quicker operation speed than the ED method. However, it has one drawback that local minima happen unexpectedly [1, 9]. Moreover, in the BMA method, a fixed camera is mainly used and the block size is also kept constant [10]. Therefore, these systematic constraints cause low object-tracking performance for mobile robots.

The purpose of this paper is to improve the tracking performance in the above-mentioned applications that use the methods of ED and BMA. The proposed method has two advantages in regards to object tracking. One is the improvement of speed during image processing and the other is improvement of object recognition ability. These enhancements can be acquired by using RB (Representative Block), especially the variable size RB that is newly proposed in this paper. Using a 2-DOF (Degree Of Freedom) active camera mounted on a mobile robot, we demonstrated that high tracking performance could be obtained with the variable size RB. The advantages of

Manuscript received October 10, 2002; revised June 10, 2003; accepted August 18, 2003. Recommended by Editorial Board member Whee-Kuk Kim under the direction of Editor Chung Choo Chung.

Wan-Cheol Kim was with Pusan National University, Department of Electronics Engineering, 30 Changjeon-dong, Kumjeong-Ku, Pusan, 609-735, Korea (e-mail: maldug2 @msn.cm).

Cheol-Ho Hwang and Prof. Jang-Myung Lee are with Pusan National University, Department of Electronics Engineering, 30 Changjeon-dong, Kumjeong-Ku, Pusan, 609-735, Korea (e-mail: {chhwang, jmlee}@pusan.ac.kr).

the variable size RB are particularly and remarkably demonstrated when the various sizes of objects exist in an image plane.

This paper is composed of the following sections: Section 2 explains the BMA method whose performance is compared with the optimal RB, and Section 3 includes the definition of the RB. The size determination of the RB is mentioned in Section 4, which defines the optimal RB. Section 5 deals with experiments using the RB and displays the results. Finally, we summarize our ideas in Section 6.

2. BLOCK MATCHING ALGORITHM

Frames n and $n-1$ are often referred to as the present frame and previous frame, respectively. In the BMA method, the present frame of the sequence is divided into rectangular or square blocks of pixels. For each block in the current frame, we look for the block of pixels that is the closest to the block in the previous frame, according to the predetermined criterion. The relation between the consecutive two corresponding blocks describes a motion vector.

The best match can be found by minimizing a distortion measurement through criterion, such as the MAD (Mean Absolute Difference) and MSD (Mean Square Difference), or by maximizing a correlation function of the two blocks. Matching of the blocks can be quantified according to various criteria including the maximum cross-correlation, the minimum MSD, the minimum MAD and the maximum MPC (Matching Pixel Count) [10, 11]. Several matching criteria for BMA are summarized below:

(a) Mean Absolute Difference (MAD)

$$MAD(u,v) = \frac{1}{N_1 \times N_2} \sum_{i=0}^{N_1-1} \sum_{j=0}^{N_2-1} \left| L_n(i+u, j+v) - L_{n-1}(i,j) \right| \tag{1}$$

(b) Mean Square Difference (MSD)

$$MSD(u,v) = \frac{1}{N_1 \times N_2} \sum_{i=0}^{N_1-1} \sum_{j=0}^{N_2-1} \left| L_n(i+u, j+v) - L_{n-1}(i,j) \right|^2 \tag{2}$$

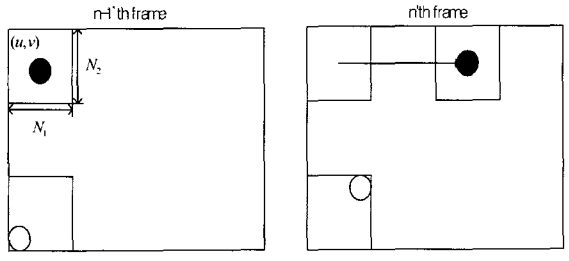


Fig. 1. Block matching algorithm.

(c) Cross-Correlation Function (CCF)

$$CCF(u,v) = \frac{\sum_{i=0}^{N_1-1} \sum_{j=0}^{N_2-1} L_n(i,j) L_{n-1}(i+u, j+v)}{\left[\sum_{i=0}^{N_1-1} \sum_{j=0}^{N_2-1} L^2{}_n(i,j) \right]^{1/2} \left[\sum_{i=0}^{N_1-1} \sum_{j=0}^{N_2-1} L^2{}_{n-1}(i+u, j+v) \right]^{1/2}} \tag{3}$$

(4) Matching Pixel Count (MPC)

$$T(u,v,i,j) = \begin{cases} 1, & if \left| L_n(i+u, j+v) - L_{n-1}(i,j) \right| \\ & \leq Threshold \\ 0, & otherwise \end{cases}$$

$$MPC(u,v) = \sum_{i=0}^{N_1-1} \sum_{j=0}^{N_2-1} T(u,v,i,j), \tag{4}$$

where L_{n-1} and L_n represent values of the pixel in $n-1$th and nth frame, respectively and (u,v) is a search point in the search area.

Among the above matching criteria, CCF and MSD require comparison and multiplication while the others require comparison and accumulation. Since multiplication generally requires a greater degree of hardware complexity than comparisons, it is more expensive to implement multiplication. MAD is most widely used due to its lower complexity, while its performance is comparable to that of MSD. Although MPC requires less hardware complexity than MAD, its performance is quite sensitive to the threshold value selected [10]. So in this paper, we use the MAD method as a comparison with RB.

In Fig. 1, the motion vector of the dark object is easily computed because the displacement of the object is bigger than the size of block. However, the motion vector of the grey object may have been lost because the object displacement is smaller than the block size. If we mainly focus on the object region, we can use the cross-correlation rather than MAD or MSD, although it is difficult for cross-correlation criteria to locate the motion vector because of the real-time heavy computational burden. Therefore, in this paper, we have defined RB (Representative Block) and applied an optimal RB to maintain the motion vector. The optimal RB has been verified in this paper to provide high performance for real-time moving object tracking.

3. REPRESENTATIVE BLOCK MODEL

The other algorithms demonstrated low performance due to the multitude of operations required to achieve accurate object recognition and

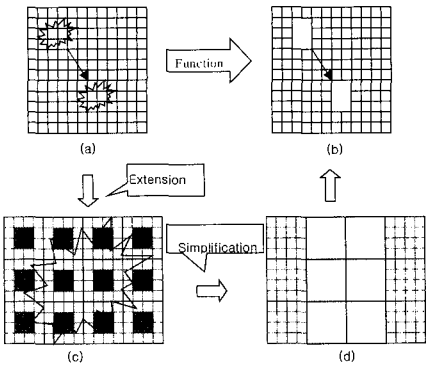


Fig. 2. Representative Block (RB).

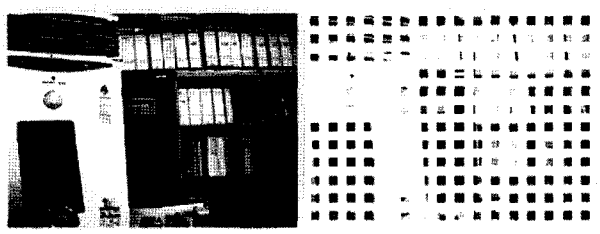
the numerous problems related to real-time object tracking. In this paper, our main interest is in moving vector elements.

The moving vector elements of an object are represented in Fig. 2 (a), which are converted to a block and shown as Fig. 2 (b). Here, detection of the movement vector through simplification is the focus. The object in Fig. 2 (a) can be distinguished from the background through gray block sampling as shown in Fig. 2 (c). Fig. 2 (d) represents the simplified object region represented as the representative blocks. On the whole, Fig. 2 (a) can be represented as Fig 2 (b). Therefore, the gray blocks are defined as RB (Representative Block) and the block (2x2) is representative of the larger block in Fig. 2 (d).

As shown in Fig. 2, when the image data are processed in each pixel unit, the dimension of image processing is increased by 16 times. Therefore, for both to decrease the dimension of data processing and to improve the image data processing speed, the RB (Representative Block) is defined. Since RB covers its neighboring pixels, the region of RB and its neighbor is defined as S_{OB} which is generally represented as,

$$S_{OB} = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & (S_{RB})_{k,k} & \cdots & (S_{RB})_{k,N-k+1} & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & (S_{RB})_{N-k+1,k} & \cdots & (S_{RB})_{N-k+1,N-k+1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad (5)$$

where S_{OB} is an $N \times N$ square matrix, $S_{OB} \in R^{N \times N}$, and sub-matrix S_{RB} is an $(N-k)$ by $(N-k)$ square matrix, $S_{RB} \in R^{(N-k)\times(N-k)}$, with a half dimension of S_{OB}, which is defined as the RB.



(a) Original image. (b) Image with the RB.

Fig. 3. Original image and image represented by the RB.

The calculation load in the image processing can be reduced since we use only the RB to track the moving object. Actually, only the RB is read from the image buffer in the vision board and the operation is adapted to the RB. Therefore, the computational cost is reduced remarkably by utilizing the RB. In the BMA method, an image is divided into those blocks and every pixel within the blocks is computed. However, when we utilize the RB, the blocks are shrunk to RB's, which reduces the computational load to a quarter.

Fig. 3 illustrates an original image and an image represented by the RB. In the BMA method, the motion vector is located by using MAD, MSD or Cross-Correlation. In this paper, when the optimal RB is used, we have used the True or False concept. True or false presents existence or nonexistence of an object in the RB, respectively. We have also assumed that the information of a moving object in uniform surroundings has been obtained *prior*. If the pixels corresponding to the object are more than 50% in the RB, it is assigned to the True and the image for the RB is displayed as black. If the pixels corresponding to the object are less than 50% in the RB, it is assigned to the False and the corresponding region is displayed as white. When a RB is assigned to the True method, it is considered as including the object's information, and we compute the center point of the object as:

$$(x_O, y_O)_{n-1} = \left(\frac{1}{m} \sum x_s, \frac{1}{m} \sum y_s \right), \quad (6)$$

where $(x_O, y_O)_{n-1}$ is the center point of the object in the $n-1$th frame and m is the number of the RB as True and (x_s, y_s) is the center point of the RB's in the True. In the same way, we find the RB's in the True and the center point of the object $(x_O, y_O)_n$ in the nth frame. After we have found the RB's as the True and the center point of the object $(x_O, y_O)_n$ in the nth frame, the motion vector is calculated simply by subtracting $(x_O, y_O)_{n-1}$ from $(x_O, y_O)_n$.

4. OPTIMAL REPRESENTATIVE BLOCK

The size of RB should be adjusted according to the size of the object in the image plane. Since the object is moving, in the image plane, the size and shape of the object can be changed due to the distance and angle between the camera and the object. Of course, the object image size becomes smaller when the object moves farther away and the image size becomes larger when the object moves closer. The shape and size of the object can be changed as the camera's angle changes. So, in this work, we selected a small ball because a ball is always shown as a circle from all directions, and we can obtain the ball size using distance information. As a result, we can predict the ball size by using perspective transformation in the kinematics of a two DOF active camera [12-14].

4.1 Perspective transformation

Perspective transformation (also called imaging transformation) projects 3-D points onto a plane. Perspective transformation plays a central role in image processing because it describes mathematically how an image is formed by viewing a 3-D world.

Fig. 4 illustrates a perspective model of the image process. The camera coordinate system (x, y, z) has the image plane coincident with the xy plane and the optical axis (established by the center of the lens) along the z axis. Thus, the center of the image plane is at the origin, and the center of the lens is at coordinates $(0, 0, \lambda)$. If the camera is in focus for a distant object, λ is the focal length of the lens. Here the assumption is that the camera coordinate system is aligned with the world coordinate system (X, Y, Z).

Let (X, Y, Z) be the world coordinates of any point in a 3-D scene, as shown in Fig. 4. Throughout the following discussion we assume that $Z > \lambda$; that is, all points of interest lie in front of the lens. The first step is to obtain a relationship that provides the coordinates (x, y) of the projection of the point (X, Y, Z) onto the image plane. This is easily accomplished by the use of similar triangular properties. With reference to Fig. 4,

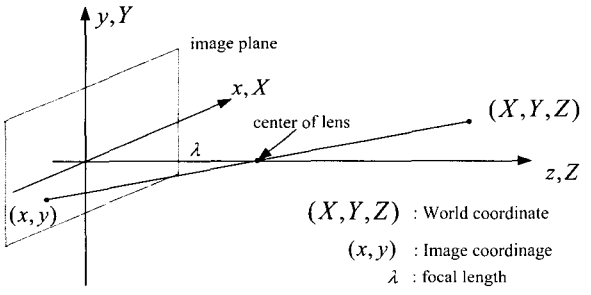


Fig. 4. Perspective model of the imaging process.

$$\frac{x}{\lambda} = -\frac{X}{Z-\lambda} = \frac{X}{\lambda - Z}, \tag{7-1}$$

$$\frac{y}{\lambda} = -\frac{Y}{Z-\lambda} = \frac{Y}{\lambda - Z}, \tag{7-2}$$

where the negative sign implies that X and Y indicate the image points, which are actually inverted. Now the image-plane coordinates of the projected 3-D point follow directly from as,

$$x = \frac{\lambda X}{\lambda - Z}, \tag{8-1}$$

$$y = \frac{\lambda Y}{\lambda - Z}. \tag{8-2}$$

4.2 Ball size

The active camera on the mobile robot is configured so that the ball information exists in the middle of the input image at all times. Therefore, as the mobile robot moves closer to the target, the adjustment of the pan and tilt angle of the active camera in the robot system enables the information of the target to be directly in the center of the input image at all times. Hence, both the ball center and the center of the camera lens are continually positioned on the Z-axis as shown in Fig. 5. Note that (X_0, Y_0) and (X_1, Y_1), determine the ball diameter. The ball size on the image frame can be changed by only the Z_0, which is the distance between the ball and the camera. First of all, we can obtain Z_{ccd} (camera height) and θ_p (tilt angle) by using the kinematics of the 2-DOF active camera mounted on the mobile robot to obtain R, the radius of the ball. The corresponding points on the image plane, (x_0, y_0) and (x_1, y_1), can be computed as follows:

$$(x_0, y_0) = \left(\frac{\lambda X_0}{\lambda - Z_0}, \frac{\lambda Y_0}{\lambda - Z_0} \right), \tag{9-1}$$

$$(x_1, y_1) = \left(\frac{\lambda X_1}{\lambda - Z_0}, \frac{\lambda Y_1}{\lambda - Z_0} \right), \tag{9-2}$$

where Z_0 is

$$Z_0 = \lambda + \frac{Z_{ccd}}{\sin\theta_p} - \frac{R}{\cos\left(\pi/2 - \theta_p\right)}. \tag{10}$$

Using the point values, (x_0, y_0) and (x_1, y_1), we can calculate the radius of the ball from the circle equation

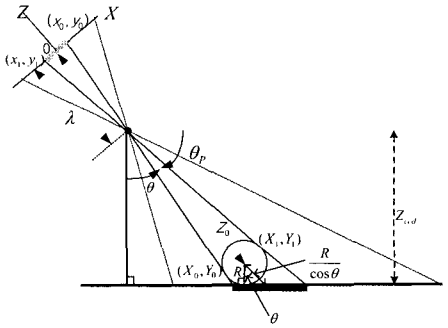


Fig. 5. Distance between robot camera and ball.

$$\left(x-\frac{x_0+x_1}{2}\right)^2+\left(y-\frac{y_0+y_1}{2}\right)^2 = \left(\frac{\sqrt{(x_0-x_1)^2+(y_0-y_1)^2}}{2}\right)^2, \quad (11)$$

$$r=\frac{\sqrt{(x_0-x_1)^2+(y_0-y_1)^2}}{2}, \quad (12)$$

where r is the radius of the ball in the image frame. We are presently going to determine the size of the RB proportional to the radius of the ball on the image plane.

4.3 Optimal representative block

The size of RB can be adjusted according to the size of the ball in the image frame, namely the distance between the ball and the camera on the robot. In this paper, the size of RB is selected so as to maintain the constraints of utilizing a 320x240 pixel image and the size of the ball in the image plane is represented by the natural number.

Fig. 6 represents the algorithm of the optimal RB. First, the size of the ball is obtained in an image plane. The optimal RB is initially determined by the radius of the ball from points through (7~12). When the ball moves, the 2-DOF active camera begins to track the center point of the ball, which can be calculated by the pre-mentioned center of the RB, to align it with the center of the camera lens by changing both pan and tilt angle. Z_0, the distance between the ball and the camera, is estimated through pan and tilt angles, and finally optimal RB is obtained by the ball's radius, which can be calculated via Z_0. Similarly, when the ball continuously moves, Z_0 is estimated by using the pan and tilt angles and then the radius of the ball is computed from Z_0. Therefore, the optimal RB can be acquired from the ball radius.

Half of the object's size, S_{OB} is approximated as

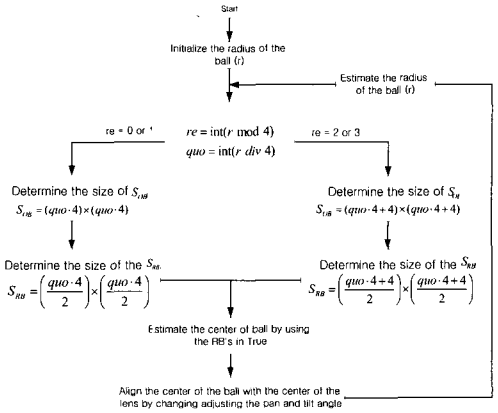
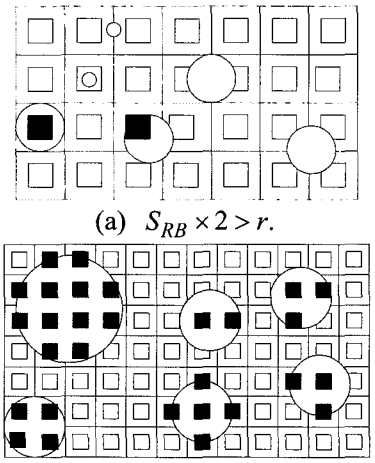


Fig. 6. Algorithm to determine the RB.



(a) $S_{RB} \times 2 > r$.

(b) $S_{RB} \times 2 \le r$.

Fig. 7. Examples with different sizes of RB's.

a square, that is, $S_{OB} = (quo \cdot 4) \times (quo \cdot 4)$. Now the size of optimal RB, S_{RB}, is decided by the two conditions so as not to lose the moving object. The first condition is that it should satisfy the constraint:

$$S_{RB} \times 2 \le r, \quad (13)$$

where S_{RB} is the size of the RB and r is the radius of the ball. Equation (13) provides the minimal size of the object to be recognizable in the image plane with the RB's.

The second condition is that the size, S_{OB}, should be the multiple of 4 in order to fit into the image frame of 320×240 pixels.

According to the first condition, Fig. 7 (a) shows the case in which the RB is too big compared to the ball such that it doesn't satisfy the condition of (13). Note that in this instance there are several occasions where the ball existence cannot be missed. Fig. 7 (b) shows the occurrence of the ball size being large enough compared to the size of RB, which satisfies

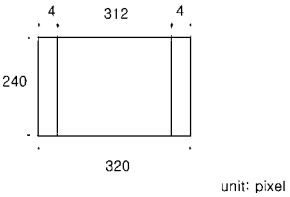


Fig. 8. When $S_{RB} = 6$, there exists uncovered region by the RB's.

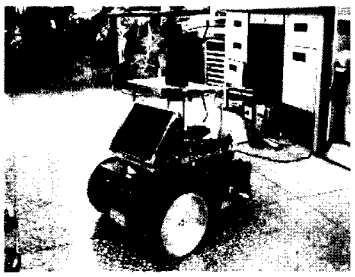


Fig. 9. Experimental mobile robot.

(13). The black squares are the RB's recognized as True in Fig. 7. If we choose the smaller RB than the half of the ball's radius, the number of the RB's in the True may increase for the ball. This may result in the increase of the computational load to calculate the center point of RB's in the True. Consequently, the optimal size of RB is chosen as the half of the ball's radius in order to minimize the computational load.

According to the second condition, the optimal RB size should be a multiple of 2 in order to satisfy (13) as well as be half of S_{OB}. So, S_{OB} should be a multiple of 4 and identical with the radius of the ball by (5) and (13). Therefore, if the ball's radius is not a multiple of 4 in the image plane, we should approximate the radius of the ball to a multiple of 4 using the remainder and the quotient, the radius dividing by 4, as illustrated in Fig. 6.

In this paper, the RB has a square matrix form and the size should be a multiple of 2 among measures of 40 that is half of the largest common measure of 320 and 240 to divide the image data in 320x240 pixels into integer numbers of blocks. Therefore, the sizes of 2, 4, 8, 10, 20 and 40 satisfy this condition. To keep the constraint, it is difficult to increase the size of RB linearly proportional to the ball size. Therefore, we may choose the RB size as a multiple of 2, including the even numbers between 2 and 40, if necessary. By this inclusion, all of the sub-blocks may not have the same size, which is illustrated by Fig. 9.

As shown in Fig. 8, for an example, when the size of RB is 6, we may assign the margins of 4 pixels to the left and the right sides. These parts can be excluded in the image processing to make the numbers of RB integer. When the size of RB is 6 (pixels), the diameter of the ball is represented in 24 pixels. Even though only 312x240 pixels are processed for the ball tracking, it does not lose track of the ball because the diameter of the ball is larger than the side margin of 4 pixels.

5. EXPERIMENTS AND RESULTS

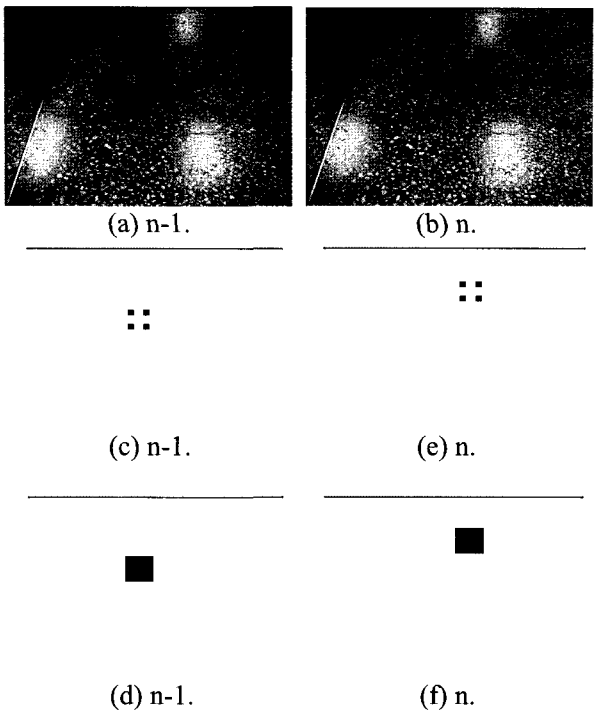
5.1 Equipment for experiments

Fig. 9 is a photograph of the mobile robot used in the actual experiment [14]. In the mobile robot, there are six controllers to control two wheels and the pan/tilt camera, to gather the gyro sensor data, and to coordinate all these controllers by one high controller that is a Pentium 3 PC board. There is also a frame grabber card to obtain the information of the image, which is interfaced through PCI bus to the PC. The CAN-bus is used for the controllers, and there is an interface card for the active camera to control through ISA bus.

5.2 Results

We performed three experiments. In the first experiment, the distance between the ball and camera is 83cm and the ball moves by 17cm in an upward shift along the corridor, which is shown in the difference between Fig. 10 (a) and Fig. 10 (b). The comparison of the optimal RB and the conventional BMA method is carried out. The result of the first experiment shown in Fig. 10 (a) is the previous image in n-1 frame and (b) is the present image in n frame. (c) and (e) are the resultant images using the optimal RB, which has 20x20 pixels after object tracking. (d) and (f) are the resultant images using the BMA method, which has 20x20 pixels after object tracking. Intentionally, the block size of BMA is kept identical to the sub-block size of the optimal RB. However, within a block of BMA all pixels (20x20x3=1200) are processed, while the optimal RB (10x10x3=300) must be processed within a sub-block. Thus, the computational load is reduced to a quarter. In other words, although both methods are satisfactory in the sense of tracking and recognition, the optimal RB is superior to the BMA in relation to operational speed. In the BMA, it elapsed about 29~31 msec, while in the optimal RB, about 18~20 msec lapse exists for object recognition.

The second experiment is executed for comparison between the optimal RB and fixed RB. Fig. 11 shows the result of the second experiment. The distance between the camera and the ball is 186cm and the radius of the ball equals 7 pixels. Here, in Fig. 11 (a), the fixed size of the RB's, which have 20x20 pixels are used. In that case, the position of the ball can't be recognized since the RB fails to notice the ball information, as shown Fig. 7 (a). Although computational load is reduced as compared with the BMA method, the moving object recognition ability

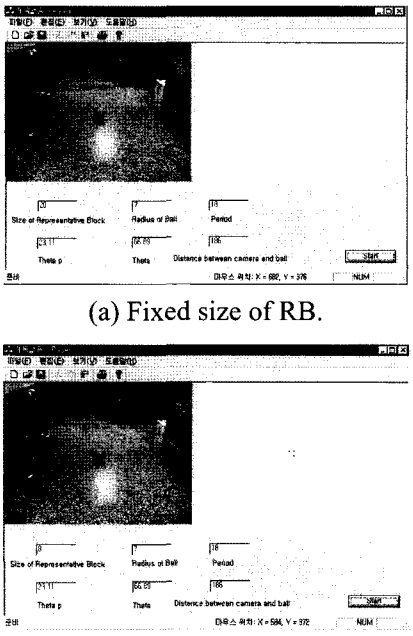


(a) n-1. (b) n.

(c) n-1. (e) n.

(d) n-1. (f) n.

Fig. 10. Comparison between the optimal RB and BMA.

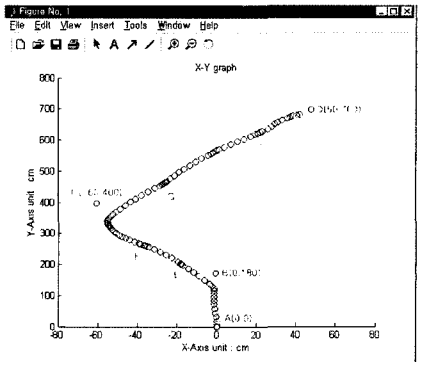


(a) Fixed size of RB.

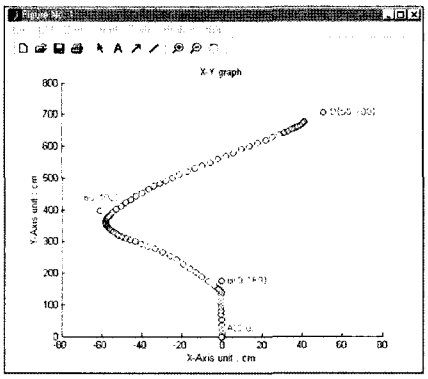
(b) Optimal RB's size (8).

Fig. 11. Comparison between a Fixed and an optimal RB.

is unsatisfactory. Fig. 11 (b) indicates the result with the optimal RB that has 8x8 pixels according to the ball size. This shows successful tracking through three occupied RB's. Not only is computational load reduced but also moving object recognition is improved. Hence, the optimal RB demonstrates very high performance in tracking of the moving object.



(a) In BMA.



(b) In RB.

Fig. 12. Tracking data with BMA or RB.

Fig. 12 shows the encoder data of the mobile robot in BMA or RB from the third experimentation results. When the object moves from A (0,0) to D (50,700) via B (0,180) and C (-60,400) with 1 (m/sec), the mobile robot tracks the moving ball using the BMA (In Fig. 12 (a)) or RB (Fig. 12 (b)). In Fig. 12 (a) because the ball displacement is smaller than the block size in BMA as mentioned in Section 2, there exists several stop points (E, F, G, H, and I), where the mobile robot halts the tracking for 2 seconds. At the stop points, although the ball moves, the mobile robot cannot recognize the movement with the BMA method. However, in Fig 12 (b), the mobile robot tracks the ball successfully with the RB method.

6. CONCLUSION

This paper proposed an optimal RB method for real time object tracking, and dealt with the comparison between RB method and BMA method with the purpose of demonstrating superiority of the RB method. In the case of BMA, image preprocessing is required for the whole block to recognize the correct image, which results in the heavy processing load and the loss of the object being tracked. Disappearance of the object also takes place when the displacement of the object is less than the block size in the image when BMA with MSD and

MAD is applied for the moving object tracking algorithm in real-time. When the fixed RB method is utilized for the tracking, sometimes the algorithm misses the moving object in the image frame, although it reduces the computational time. However, when the optimal RB is utilized for the tracking, the effects of both high tracking performance and decreased computational time are obtained concurrently. For the decision of the RB size, the size of the tracking object in the image frame is the major factor. Therefore, this RB method with the size optimization scheme can be widely used for the applications that include the tracking of fast moving objects in real-time situations.

REFERENCES

[1] B. Liu and A. Zaccarin, "New Fast Algorithms for the Estimation of Block Motion Vectors," *IEEE Trans. Circuits*, vol. 3, no. 2, pp. 148-157, Apr. 1993.

[2] J. S. Kim and R. H. Park, "A Fast Feature-Based Block Matching Algorithm Using Integral Projections," *IEEE Journal on Selected Area in Communication*, vol. 10, no. 5, pp. 968-971, Jun. 1992.

[3] Y. C. Lin and S. C. Tai, "Fast Full-Search Block-Matching Algorithm for Motion-Compensated Video Compression," *IEEE Trans. on Communications,* vol. 45, no. 5, May 1997.

[4] W. J. Hwang, Y. C. Lu, and Y. C. Zeng, "Fast Block-Matching Algorithm for Video Coding," *Electronics Letters*, vol. 33, no. 10, May 8, 1997.

[5] R. Canals, A. Roussel, J. L. Famechon, and S. Treuillet, "A Biproce ssor-Oriented Vision-Based Tracking System," *IEEE Trans. Industrial Electronics*, vol. 49, no. 2, Apr. 2002.

[6] M. B. Ahmad, D. Y. Kim, K.S. Roh, and T.S. Choi, "Motion Vector Estima tion Using Edge Oriented Block Matching Algorithm for Video Sequences," *Image Processing*, 2000. Proceedings. 2000 Interna tional Conference, vol. 1, pp. 806-863, 2000.

[7] M. Brünig and W. Niehsen, "Fast Full-Search Block Matching," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 11, no. 2, Feb. 2001.

[8] Y. H. Fok and O. C. Au, "An Improved Fast Feature-Based Block Motion Estimation," *Image Processing, IEEE International Conference*, vol. 3, pp. 741-745, Nov. 1994.

[9] T. Zahariadis and D. Kalivas, "Fast algorithms for the estimation of block motion vectors," *ICECS*, pp. 716-719, 1996.

[10] D. K. Lim and Y. S. Ho, "A Fast Block Matching Motion Estimation Algorithm based on Statistical Properties of Object Displacement," *TENCON '98. 1998 IEEE Region 10 International Conference on Global Connectivity in Energy, Computer, Communi cation and Control*, vol. 1, pp. 138-141, 1998.

[11] M. J. Chen, L. Gee, T. D. Chiueh, and Y.P. Lee, "A New Block-Matching Criterion for Motion Estimation and its Implementation," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 5, no. 3, Jun. 1995.

[12] Y. Zhao and S. L. BeMent, "Kinematics, Dynamics and Control of Wheeled Mobile Robots," *IEEE Conf. Robotics and Automation*, pp. 91-96, 5. 1992.

[13] D. J. Kreigman, E. Triendl, and T. O. Binford, "Stereo vision and navigation in building for mobile robots," *IEEE Trans. on Robotics and Automation*, vol. Ra-2, pp. 14-23, 1986.

[14] I. M. Kim, W. C. Kim, K. S. Yun, and J. M Lee, "Navigation of a Mobile Robot Using Hand Gesture Recognition," *Journal of Control, Automation and Systems Engineering*, vol. 8, no. 7, pp. 599-606, July, 2002.

Wan-Cheol Kim received the B.S. degree in the Department of Electrical Engineering from Pusan National University in 2003. His research interests include robot vision, navigation, and image processing.

Cheol-Ho Hwang received the B.S. degree in the Department of Electrical Engineering from Pusan National University in 2003. His research interests include mobile robot, navigation, and GPS.

Jang-Myung Lee has been a Professor in the Department of Electronics Engineering, Pusan National University since 1992, where he is currently Director of the Research Institute for Computer Information and Communication. He received the B.S. and M.S. degrees in Electronics Engineering from Seoul National University in 1980 and 1982, respectively and the Ph.D. degree in Computer Engineering from the University of Southern California in 1990. His current research interests include intelligent robotic systems, integrated manufacturing systems, cooperative control and sensor fusion. Dr. Lee is an IEEE Senior member, and a member of ICASE and IEEK. (http://robotics.ee.pusan.ac.kr)