PAPER

Exclusive Block Matching for Moving Object Extraction and Tracking

Zhu LI[†a], **Kenichi YABUTA**[†b], ***and*** **Hitoshi KITAZAWA**[†c], ***Members***

SUMMARY Robust object tracking is required by many vision applications, and it will be useful for the motion analysis of moving object if we can not only track the object, but also make clear the corresponding relation of each part between consecutive frames. For this purpose, we propose a new method for moving object extraction and tracking based on the exclusive block matching. We build a cost matrix consisting of the similarities between the current frame's and the previous frame's blocks and obtain the corresponding relation by solving one-to-one matching as linear assignment problem. In addition, we can track the trajectory of occluded blocks by dealing with multi-frames simultaneously.
key words: *moving object extraction, tracking, block matching, linear assignment*

1. Introduction

Tracking objects is an elementary task in online and offline image-based applications including traffic surveillance, motion capture, and robot vision, etc. After years of researches, many efficient methods have been proposed. However, in order to achieve accurate motion analysis of moving object, it is necessary to obtain the corresponding relation of each part in an object between consecutive frames.

Background subtraction [1], which is one of the commonly used techniques, detect moving objects on the pixel level and can not provide the corresponding relation of each part. Mean-Shift [2] and Particle Filter [3], [4] which are very popular for object tracking in present research use the global information of object. They track moving object on object-level and do not obtain the corresponding relation of each part.

On the other hand, optical flow methods such as Block Matching, Horn-Schunck method [9], Lucas-Kanade method [10] and a method based on SIFT descriptor [5] estimate pixel motion between two frames. These methods can obtain the corresponding relations on the level of pixel by optical flow methods. However, it is impossible to extract feature points in such an area where intensity changes smoothly by the Lucas-Kanade method. A stable keypoint of SIFT descriptor is robust for illumination change, scale variance and rotation. Accurate correspondences between consecutive frames can be obtained by matching the stable keypoints. Unfortunately, on low-texture areas, stable keypoint candidates are too few to extract and track an object. Moreover, Horn-Schunck method and Lucas-Kanade method are not applicable to frames where motions of objects are very large. Motion analysis for such image sequences is still necessary.

In this paper, we propose a new method to track moving object on block-level. In order to avoid the situation that destinations of matched blocks are too close or overlap, we assume block matches in such a way that destinations are mutually exclusive and propose a method to obtain the optimal matching using linear assignment. Different from the algorithms based on graph matching [6], [7], our method simply performs block matching which does not require the graph structures of moving objects. Moreover, compared with the features of nodes and edges of graph, the features of small blocks show higher robustness under some situations such as view point change caused by object motion and illumination change. A method based on assignment, called SoftAssign, has been proposed in [8]. This method obtains a matching between 2 point sets under an affine transformation. Different from such an approximate method, the goal of our proposed method using color information is to obtain the exact solution of matching between consecutive frames which include plural objects. As shown in Fig. 1, the proposed method aims to achieve spatio-temporal continual object tracking even in the case of large motion, occlusion or shape change.

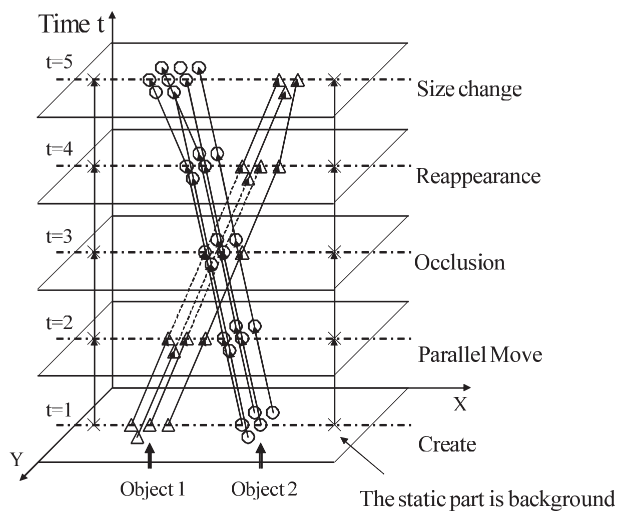


Fig. 1 Spatio-temporal tracking.

Manuscript received June 18, 2009.
Manuscript revised January 14, 2010.
[†]The authors are with the Tokyo University of Agriculture and Technology, Koganei-shi, 184–8588 Japan.
a) E-mail: lizhu@m.ieice.org
b) E-mail: kyabuta@m.ieice.org
c) E-mail: kitazawa@cc.tuat.ac.jp
DOI: 10.1587/transinf.E93.D.1263

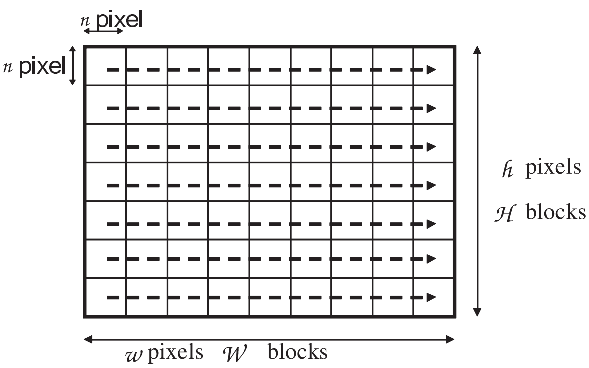


Fig. 2 Scan an image into 1 dimensional data.

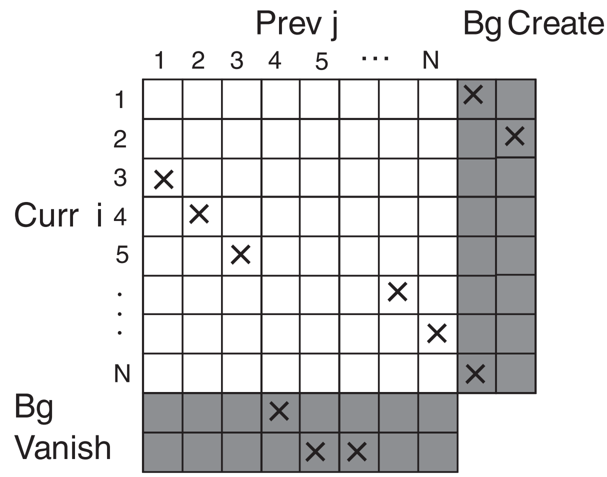


Fig. 3 Expand the matrix considering background, occlusion, creation, and vanishment.

2. Exclusive Block Matching

2.1 The Matching between Current Frame and Previous Frame

In this section we describe the basic method of block matching. Firstly, we scan the input images by block to convert the images into 1 dimensional data. If we assume that the block size is $n \times n$ pixels, the width and the height of the image are w and h, respectively, the number of blocks N is given by the equation $N = w/m \times h/n = W \times H$ as shown in Fig. 2. We build an $N \times N$ array consisting of the similarities (actually difference measure or distance) between the current frame's (Curr) blocks and the previous frame's (Prev) blocks. It is required to perform all Curr blocks by assigning exactly one Prev block to each Curr block in such a way that the total cost of the assignment is minimized. Then this problem could be solved as a linear assignment problem.

However, it is impossible to archive the one-to-one assigning as long as the object moves in the scene. This is caused by various situations involving hiddenness and reappearance of background, occlusion, creation and vanishment of moving object. Therefore, this problem can not be simply solved as a linear assignment problem.

2.2 Expand the Matrix Considering Background, Occlusion, Creation and Vanishment

In order to solve this problem, we expand the basic matrix by adding 2 rows and 2 columns as shown in Fig. 3. The columns correspond to appearance of background and creation of new blocks and the rows correspond to hiddenness of background and vanishment of blocks. We calculated the distances between current block and prevous blocks to decide which column's block is the best match. If the distance between current frame's block and background's block is closer than the distance between current frame's and the previous frame's block, this block is matched with the Bg column. If neither previous frame nor Bg column can be matched, this block is matched with the Create column. In the same way, the blocks of previous frame which match neither the current frame's nor the Bg's blocks are matched with the Vanish row. This matching problem becomes the following linear programming problem:

Minimize

$$z = \sum_{i=1}^{N+2} \sum_{j=1}^{N+2} p_{ij} c_{ij}$$

subject to

$$\sum_{j=1}^{N+2} p_{ij} = 1 \quad i = \{1, 2, \ldots, N\},$$

$$\sum_{i=1}^{N+2} p_{ij} = 1 \quad j = \{1, 2, \ldots, N\},$$

$$where\ p_{ij} = \{0, 1\} \quad \{i, j\} = \{1, 2, \ldots, N+2\},$$

$$p_{ij} = \{0\}$$

$$\{j\} = \{N+1, N+2\} \quad \{i\} = \{N+1, N+2\},$$

$$c_{ij} = \begin{cases} dist\{Curr_i, Prev_j\} \\ \quad i = \{1, \ldots, N\}, j = \{1, \ldots, N\}, \\ dist\{Curr_i, Bg_j\} \\ \quad i = \{1, \ldots, N\}, j = N+1, \\ dist\{Bg_i, Prev_j\} \\ \quad i = N+1, j = \{1, \ldots, N\}, \\ penalty\ for\ creating \\ \quad i = \{1, \ldots, N\}, j = N+2, \\ penalty\ for\ vanishing \\ \quad i = N+2, j = \{1, \ldots, N\}. \end{cases}$$

$dist\{X, Y\}$**:** distance between block X and block Y.

*penalty for creating***:** fixed value chosen when there is no block similar with it. If this value is chosen, this block is regarded as creating of a new block such as block 2 in the current frame in Fig. 3.

*penalty for vanishing***:** fixed value chosen when there is no block similar with it. If this value is chosen, this block is regarded as vanishing of a block or occlusion such as block 5 in the previous frame in Fig. 3.

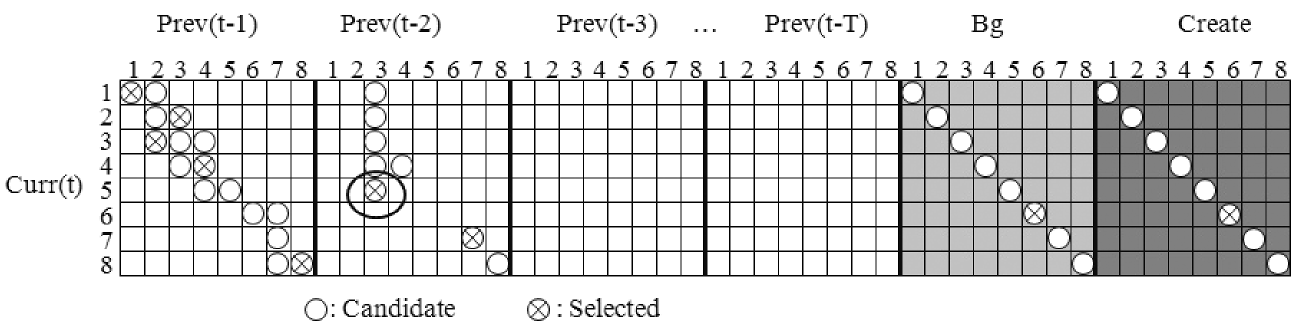


Fig. 5 Expand the cost matrix dealing with plural previous frames.

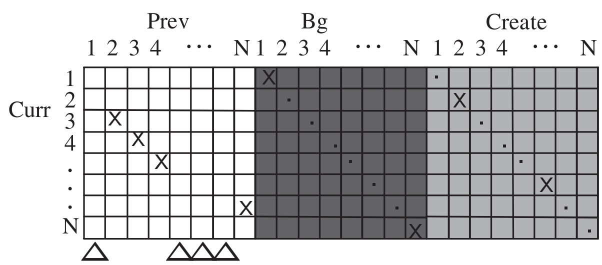


Fig. 4 Assignment problem considering background and creation.

The part which consists of the $N \times N$ array is an exclusive assignment problem. However, the part which is composed of the Bg row, Bg column, Create row and Vanish column is a partial assignment problem which is likely to be solved out with plural choices. There is no guarantee that such a problem can be solved out in a short time. For this reasaon, we ignore Bg and Vanish row to convert it to a linear assignment problem which is shown in Fig. 4. Only the diagonal elements can be selected in the part of Bg and Create. Since every row is assigned to exactly one column, only N columns are selected. Although the Bg and Vanish row are taken out, the blocks in previous frame which do not match with the blocks in the current frame (shown by △ in Fig. 4) are regarded as matched with Bg or Vanish.

2.3 Multi-Frame Expanding

Although we can discern the movement of each block by matching between current frame's and previous frame's blocks, it is still impossible to discern some situations just like occlusion and reappearance. So we expand the matrix again by dealing with plural previous frames, as shown in Fig. 5. In this figure, the number of frames T and the number of blocks N are 4 and 8, respectively. For example, block 5 in the current frame is matched with block 3 in the frame at the moment t-2. This block is regarded as occluded at the moment t-1. The value in ( ) denotes the time of the frame. The arithmetic expression is written down as follows,

Minimize

$$z = \sum_{i=1}^{N} \sum_{j=1}^{(T+2)\times N} p_{ij} c_{ij},$$

subject to

$$\sum_{j=1}^{(T+2)\times N} p_{ij} = 1 \quad i = \{1, 2, \ldots, N\},$$

$$\sum_{i=1}^{N} p_{ij} \leq 1 \quad j = \{1, 2, \ldots, (T+2)\times N\},$$

$$p_{ij} = \{0, 1\},$$

$$c_{ij} = \begin{cases} dist\{Curr(t)_i, Prev(t-T)_j\} \\ \quad i = \{1, \ldots, N\}, \\ \quad j = \{(T-1)\times N + 1, \ldots, T \times N\} \\ dist\{Curr_i, Bg_j\} \\ \quad i = \{1, \ldots, N\}, \\ \quad j = \{T \times N + 1, \ldots, (T+1)\times N\} \\ penalty\ for\ creating \\ \quad i = \{1, \ldots, N\}, \\ \quad j = \{(T+1)\times N + 1, \ldots, (T+2)\times N\}. \end{cases}$$

This is a kind of linear assignment problem and can be solved by the Hungarian method [13].

2.4 Similarity Measure

Beacuse the capability of matching is depend heavily on the calculation method of similarity, it is necessary to choose an appropriate measure. Through comparison of various experimental results, we adopt the Bhattacharyya coefficient [3] defining a distance on HSV histograms to measure the similarity between 2 blocks. The mathematical formulation of this measure is given by Eq. (1) and Eq. (2), where p and q represent 2 normalized HSV histograms.

$$\rho[p, q] = \sum_{u=1}^{m} \sqrt{p^{(u)} q^{(u)}} \tag{1}$$

$$d = \sqrt{1 - \rho[p, q]} \tag{2}$$

The HSV histogram is composed of $m = N_h N_s + N_v$ bins and we set N_h, N_s, and N_v to 10. So the m becomes 110 [3].

3. Restriction of Block State Transition

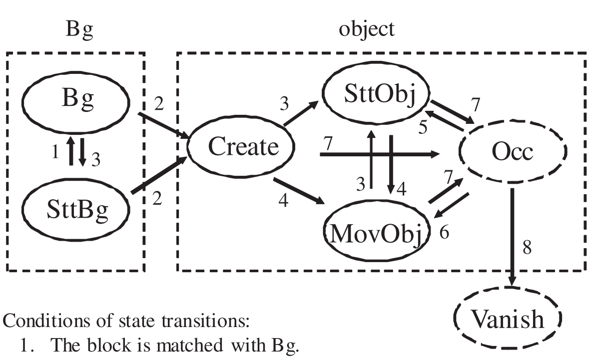
In this section, we describe the state transition of blocks. We also give an example to explain how the restriction of block state transition works on multi-frame stage.

3.1 Block State Transition

Now, we consider the state transition of blocks. Firstly, let's assume that there is a frame without any object. In our system, such frames are used as Background. Now a moving object enters the scene in a frame. In this frame, the moving object actually appears as some new blocks. From the next frame, the object will become one of several situations detailed below. If the object moves to another position, it is regarded as a moving object. If it remains on the same position, it is regarded as a static object. On condition that the object is not in the frame, it may be occluded or move out.

According to this course of object, 7 kinds of states are defined. They are MovingObject (MovObj), StaticObject (SttObj), Create, Background (Bg), StaticBg (SttBg), Occlusion (Occ) and Vanish. The state of a block can not transit to another state arbitrarily. For example, the state Bg can not transit to MovObj or SttBg before it transit to Create. Figure 6 shows the diagram of allowable block state transition.

The states of all blocks in the first frame are initially set to Bg. If a block in the current frame is matched with the Create part of the matrix, this block's state will be set to Create which means the block belongs to an object. When it is matched with the next frame, if the block moves to another position, the state will be classified as MovObj. If it is matched with the block which is on the same position, the state is classified as SttObj. On condition that none of the blocks in the scene are matched with it, the state is set to Occ. If it can not be matched with MovObj or SttObj again within a pre-assigned number of frames, it is classified as Vanish. Besides, the blocks which are matched with Bg part are classified as Bg. Finally, if the block does not belong to any object and matched with the block on the same position, it will be classified as SttBg.



Conditions of state transitions:
1. The block is matched with Bg.
2. The block is matched with Create.
3. The block is matched with Prev(t-1) on the same position.
4. The block is matched with Prev(t-1) on the different position.
5. The block is matched with Prev(t-T) on the same position. (T>1)
6. The block is matched with Prev(t-T) on the different position. (T>1)
7. The block can not be matched with any block in the scene.
8. The block can not be matched with any block within a pre-assigned number of frames .

Fig. 6 Restriction of block state transition.

3.2 Restriction of State Transition by Matrix Formulation

Now we go back to our matrix according to state transition of block shown in Fig. 6. In our matrix formulation, all rules described in previous subsection could be concentrated into 2 points. If the state of a block in previous frame is Bg or SttBg, the block could not move, which means it could only be matched with the block on the same position. If the state is Create, MovObj or SttObj, it is allowed to be matched with blocks on other positions. Considering Occ and Create could occur at any moment, all blocks in Bg columns and Create columns are always available to be matched. Here, we actually add one more constraint to confine the available range of movement of block. Let $(2d+1)\times(2d+1)$ blocks be the available range of movement around it between 2 consecutive frames. The parameter d is the available distance of movement on x and y direction. Assume a block in frame (t-T) is matched with a block in frame (t). In other words, this block is regarded as Occ from frame (t-T+1) to frame (t-1). Here T is bigger than 1. Occ only means that we do not know the exact destination of this block in these frames, and it still could move. Then the available range of movement in frame t is $(2Td + 1) \times (2Td + 1)$ blocks.

We replace the distance with a very large value in the part of the matrix which is out of the available range of movement. This processing avoids a large amount of unnecessary calculations (includes the calculation of similarities) and improves the stability of block matching. It also makes the matrix become a sparse matrix, so sparse-matrix calculation is adopted to improve the calculation speed. We will give an example of improvement of processing speed and accuracy in the section of experimental results.

3.3 An Example of Matrix Formulation Realizing the Restriction of State Transition

Figure 5 gives an example of matrix formulation which realizes the restriction of state transition. In this example, the available range of movement is 1 block. We mark the range of movement with circles. The current matching result is marked by crosses. In order to make it easy to understand, we assume that there is only 1 row with 8 blocks in each frame. Figure 7 shows the trajectories of blocks on plane. Below, we use B(t,n) and P(t,m) to represent the block n and the position m in the frame t, respectively.

The state of B(t-1,2) is a SttObj. The available range of

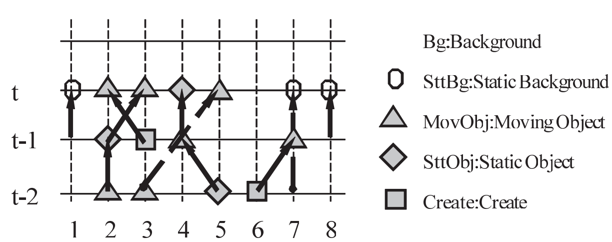


Fig. 7 States of blocks.

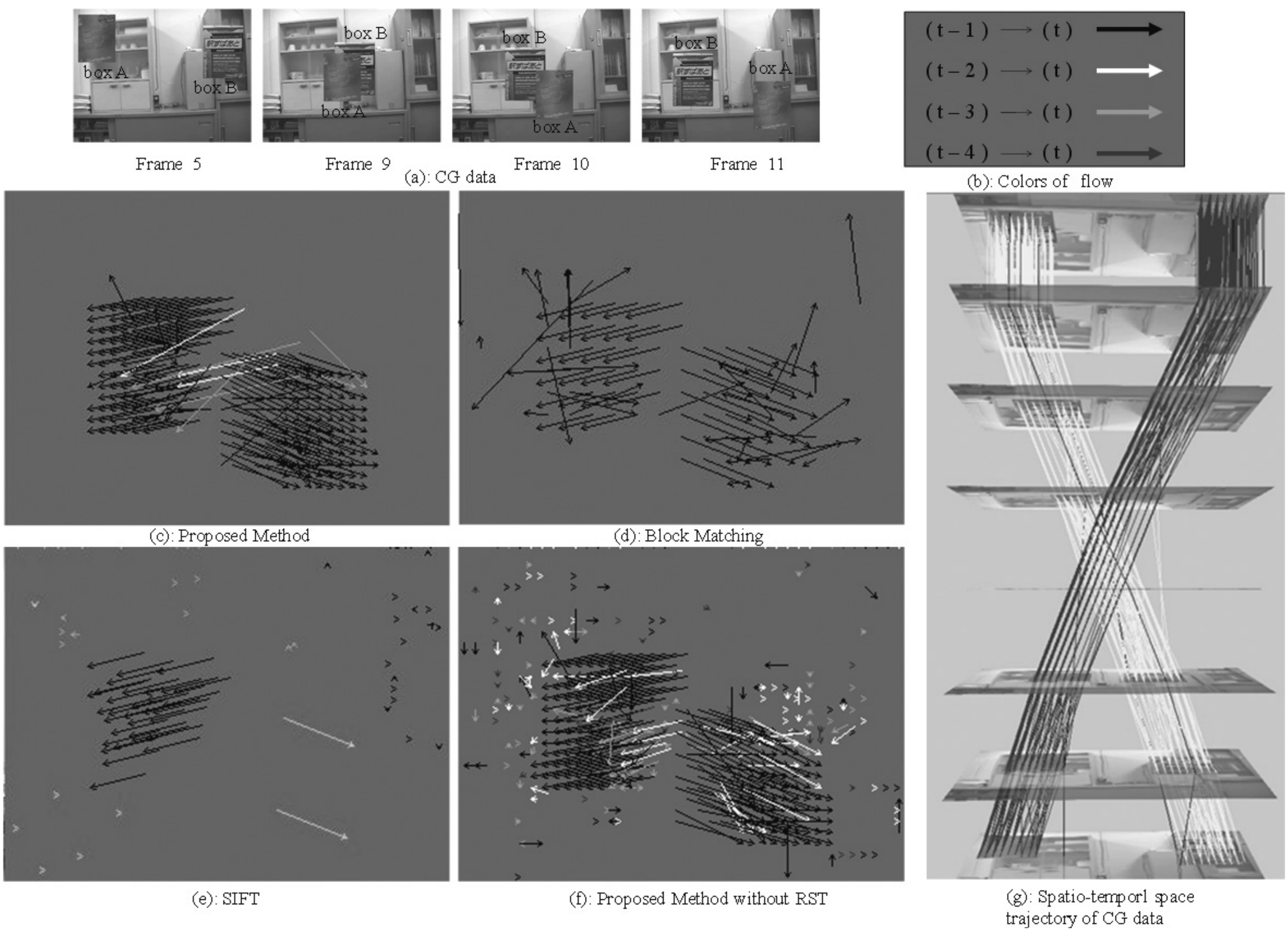


Fig. 8 The comparison between proposed method and conventional optical flow methods.

movement of this block consists of P(t,1), P(t,2) and P(t,3). In other words, it can only be matched with blocks on these positions. Since the states of B(t-1,5) and B(t-1,6) are Bg, they can only be matched with the blocks on the same position in frame (t). B(t-2,5) is matched with block on (or has moved to) P(t-1,4). This means the object is no longer on the original position, therefore B(t-2,5) will not be matched with any block in frame (t) when frame (t) becomes the current frame. B(t-2,3) is regarded as Occ in frame (t-1). The available range of movement in frame (t) thus becomes P(t,1), P(t,2), P(t,3), P(t,4) and P(t,5).

4. Experimental Results

In our experiments, the sizes of images and blocks are 320×240 pixels and 8×8 pixels, respectively. The number of previous frames is 4. So size of the cost matrix becomes 1200×7200. The available range of movement of each block is limited to $25{\times}25$ blocks around it. We normalize the Bhattacharyya distance in the range from 0 to 1000. The threshold value of creating is set to 400. In fact, if the threshold is set in the range of 300-500, experimental results are almost the same. According to our experimental results, extractions in different scenes have almost no dependence on this threshold value.

4.1 Comparison with Block Matching and SIFT

In order to prove the validity of our method, we show the experimental results of CG generated data. Two boxes enter the scene at the moment of frame 5, then move parallelly and occlusion occurs at the moment of frames 9 and 10. Figures 8 (c), (d) and (e) show the experimental results of our proposed method, block matching method, and the method based on SIFT descriptor. The optical fow is calculated by OpenCV library [11], and the code of SIFT keypoint detector is from Rob Hess's homepage [12]. We adjust the values and the numbers of parameters of optical flow methods to obtain the best results we can get. We use different lines which are shown in Fig. 8 (b) to indicate which time of frame the block is matched. Experimental results indicate that block matching method induce unnatural flows because of pushing and entering of background. The robustness of SIFT descriptor is very high, but stable keypoints are too few to extract and track box A. On the other hand, our proposed method avoids well these 2 problems, and obtained dense flows are almost parallel. Table 1 shows accuracy rates of which is denoted as AR in the table is calculated as follow,

Table 1 Accuracy rates (AR) of 4 methods. The numbers in the brackets mean (the number of accurate flows/the number of extracted flows).

Method	Proposed method	Block matching	Proposed method without RST	SIFT
AR of Box A	81.2% (69/85)	50.0% (18/36)	52.3% (46/88)	100% (2/2)
AR of Box B	84.9% (79/93)	59.5% (22/37)	58.1% (61/105)	100% (29/29)
Inaccurate flows in the background	0	6	147	1

Table 2 The improvement of calculation speed by restriction of state transition (RST).

Frame Number	5	6	7	8	9	10	11
Without RST[sec]	47.5	51.8	44.4	42.8	39.4	39.3	41.3
With RST[sec]	0.12	0.36	0.49	0.66	0.78	0.85	0.27

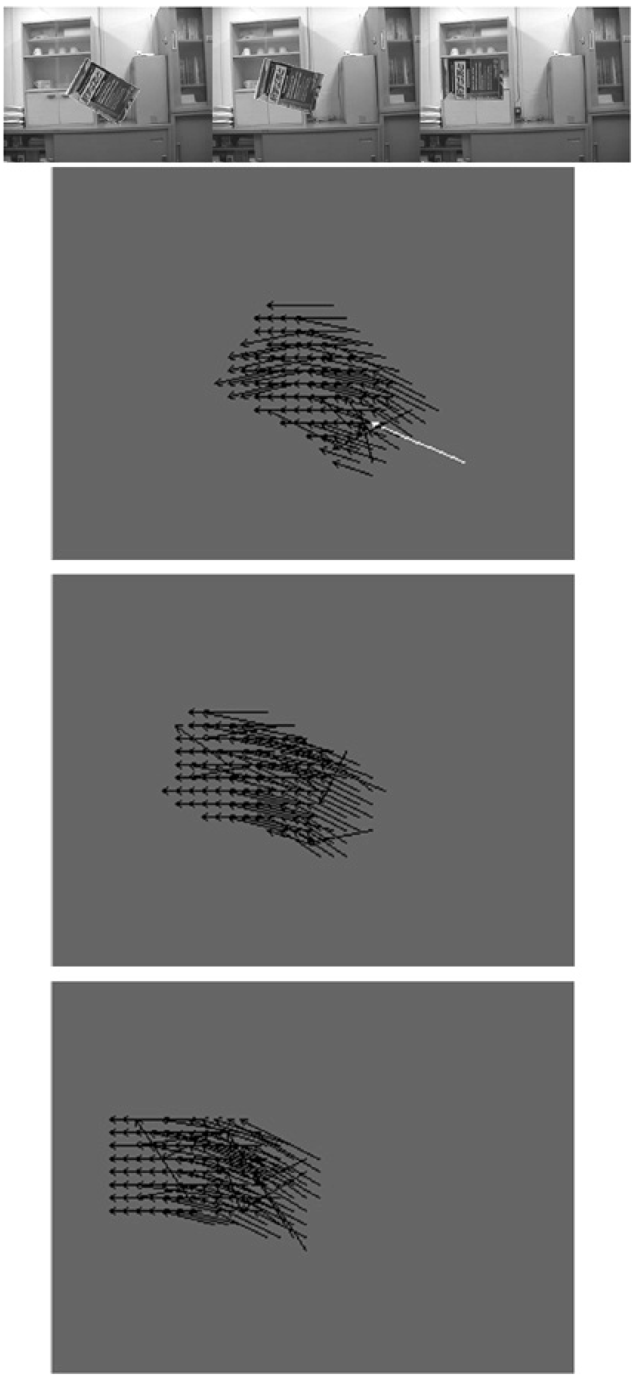


Fig. 9 Rotation.

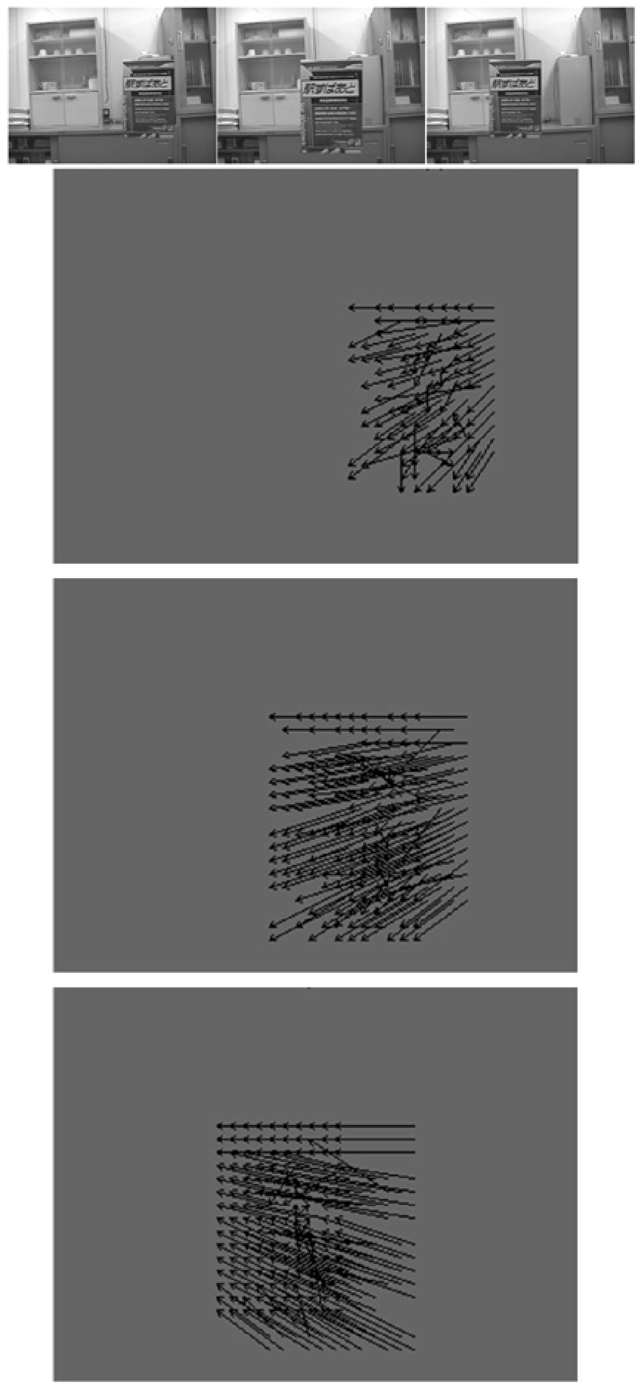


Fig. 10 Magnification.

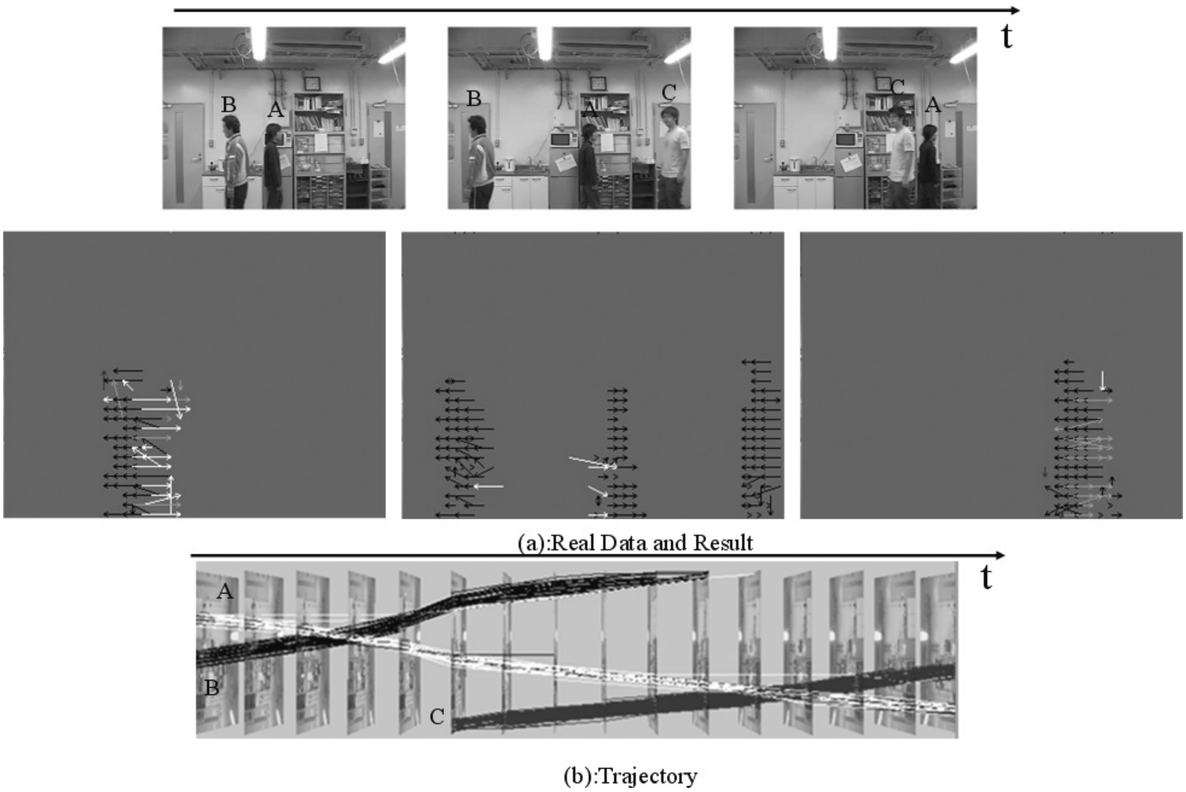


Fig. 11 Real data with occlussions.

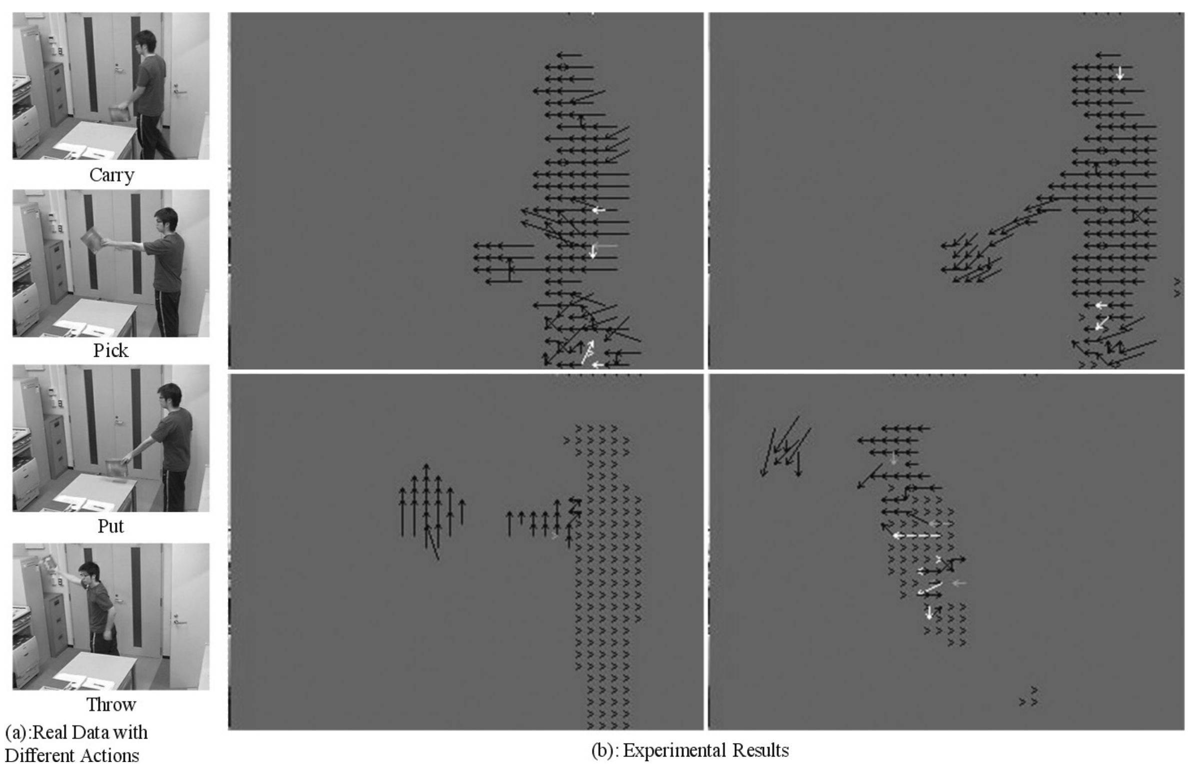


Fig. 12 Real data with different actions.

$$AR = \frac{\textit{accurate flows in the object}}{\textit{extracted flows in the object}} \quad (3)$$

4.2 Comparison with Proposed Method without RST

Figure 8 (f) shows the result without considering the restric-

Table 3 Accuracy rates of magification and rotation.

Data	Magnification	Rotation
Frame 1	73.3% (66/90)	90.3% (84/93)
Frame 2	78.1% (114/146)	85.1% (80/94)
Frame 3	81.2% (121/149)	84.4% (76/90)

tion of state transition (RST). Compared to Fig. 8 (c), the error matching derived by the previous frames is reduced by introducing RST. Moreover, the restriction of state transition can improve the calculation speed significantly. In our experiments, we use a Core 2 Duo 3.00 GHz PC with 2 G RAM under Windows XP. The calculation speed is given by Table 2. According to the result, the improvement is obvious.

4.3 Magnification and Rotation

Although we assume blocks match exclusively and do not consider the situation that a block breaks into two blocks, our method considering vanishing and creating is applicable for magnification and rotation as well. Figures 9 and 10 show the experimental results of CG data. Accurate tracking was almost realized. However, as the boundaries of blocks change, the similarities between some blocks in current frame and in frame (t-2) become higher than them between blocks in current frame and previous frame (t-1). This causes the error matching represented as white flows which appear when occlusion does not occur. Accuracy rates of magnification and rotation are given in the Tabel 3.

4.4 CG Data with Occlusion

The next example shown in Fig. 8 (g) is the trajectory of two boxes under occlusion in spatio-temporal space. We use the same data which is shown in Fig. 8 (a). The excellent result shows that the continuous tracking is not interrupted even under occlusion. In addition, flows are almost parallel that means the corresponding relation of each part between consecutive frames is obtained accurately.

4.5 Real Data

Finally, we give 2 results of real data in which shapes of moving objects are irregular. Figure 11 gives a result of real data with occlusion. Trajectory of person C is obtained accurately. Tracking of person A and B fails after occlusion. Flows of blocks on the parts of which color is very similar (clothes and hair of B and A) mixed together. Figure 12 shows another real data with different actions of a person. Althrough there are some inaccurate flows and miss extraction, the obtained flow shows the motion of each part of the person and it will be useful for motion analysis.

5. Conclusions

This paper has proposed a new method for moving object extraction and tracking based on exclusive block matching. This method has been successfully used to track moving object even in the case of occlusion and provides the corresponding relation of each part between consecutive frames.

The method can not avoid the aperture problem when the colors of blocks are same or very similar. We, therefore, can make clear that tracking is limited if we merely use color information to calculate similarity. Our future work should focus on improving the approach considering shape similarity and connectivity in consecutive blocks.

References

[1] Stauffer and W.E.L. Grimson, "Adaptive background mixture models for real-time tracking," CVPR99, Fort Colins. CO, 2246, June 1999.
[2] D. Comaniciu and P. Meer, "A robust approach toward feature space analysis," IEEE Trans. Pattern Anal. Mach. Intell., vol.24. no.5, pp.603–619, May 2002.
[3] P. Perez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," ECCV 2002, LNCS 2350, p.661, 2002.
[4] K. Nummiaro, Esther Koller-Meier, and L.V. Gool, "A color-based particle filter," Image Vis. Comput., 2002.
[5] Lowe, David G, "Object recognition from local scale-invariant features," Proc. International Conference on Computer Vision 2, pp.1150–1157, DOI:10.1109/ICCV, 1999.
[6] T.S. Caetano, C. Li, Q.V. Le, and A.J. Smola, "Learning graph matching," ICCV 2007, IEEE 11th International Conference on Computer Vision, pp.1–8, Oct. 2007.
[7] H. Jiang, M.S. Drew, and Z.-N. Li, "Linear programming matching and appearance-adaptive object tracking," EMMCVPR 2005, St. Augustine FL, USA, Nov. 2005.
[8] S. Gold, A. Rangarajan, C.-P. Lu, S. Pappu, and E. Mjolsness, "New algorithms for 2D and 3D point matching: Pose estimation and correspondence," Pattern Recognit., vol.31, no.8, pp.1019–1031, Aug. 1998.
[9] B.K.P. Horn and B.G. Schunck, "Determining optical flow," Artif. Intell., vol.17, pp.185–203, 1981.
[10] J.-Y. Bouguet, "Pyramidal implementation of the lucas Kanade feature tracker," The paper is included into OpenCV distribution.
[11] OpenCV. http://opencv.jp/document/opencvref_cv.html
[12] Code of SIFT Keypoint Detector. http://web.engr.oregonstate.edu/hess/index.html
[13] Munkres assignment algorithm. http://csclab.murraystate.edu/bob.pilgrim/445/munkres.html
[14] M.A. Trick, "A linear relaxation heuristic for the generalized assignment problem," Naval Research Logistics, vol.39, pp.137–152, 1992.

Zhu Li received the B.S degree in electrical and information engineering from ZheJiang University of Technology, China, in 2005. Then he received M.S degree in electrical and electronic engineering from Tokyo University of Agriculture and Technology, Japan, in 2009. He is currently a Ph.D. student of Department of Electronic and Information Engineering at the Tokyo University of Agriculture and Technology. His research interests are in Image Processing and Object detection.

Kenichi Yabuta received the B.S and the M.S degrees in electrical and electronic engineering from Tokyo University of Agriculture and Technology, Japan, in 2005 and 2007, respectively. He is currently a Ph.D. student of Department of Electronic and Information Engineering at the Tokyo University of Agriculture and Technology. His research interests are in Image Processing and Optimizing Algorithms.

Hitoshi Kitazawa received his B.S, M.S and Ph.D. degrees in electronic engineering from Tokyo Institute of Technology, Tokyo Japan, in 1974, 1976 and 1979, respectively. He joined the Electrical Communication Laboratories, Nippon Telegraph and Telephone Corporation (NTT), in 1979. Since 2002, he is a professor at Tokyo University of Agriculture and Technology. His research interests are in VLSI CAD algorithm, Computer Graphics and Image Processing. He is a member of IPSJ and IEEE.