Article Information

Title	Extraction and Tracking Moving Objects in Detail Considering Visual Feature Constraint and Structure Constraint
Authors	Zhu LI, Yoichi TOMIOKA, and Hitoshi KITAZAWA
Citation	IEICE TRANSACTIONS on Information and Systems, Vol.E96-D, No.5, pp.1171-1181
Copyright	
IEICE Transactions Online URL	https://search.ieice.org/

PAPER

Extraction and Tracking Moving Objects in Detail Considering Visual Feature Constraint and Structure Constraint*

Zhu LI[†a)], **Yoichi TOMIOKA**[††b)], ***and*** **Hitoshi KITAZAWA**[††c)], ***Members***

SUMMARY Detailed tracking is required for many vision applications. A visual feature-based constraint underlies most conventional motion estimation methods. For example, optical flow methods assume that the brightness of each pixel is constant in two consecutive frames. However, it is difficult to realize accurate extraction and tracking using only visual feature information, because viewpoint changes and inconsistent illumination cause the visual features of some regions of objects to appear different in consecutive frames. A structure-based constraint of objects is also necessary for tracking. In the proposed method, both visual feature matching and structure matching are formulated as a linear assignment problem and then integrated.

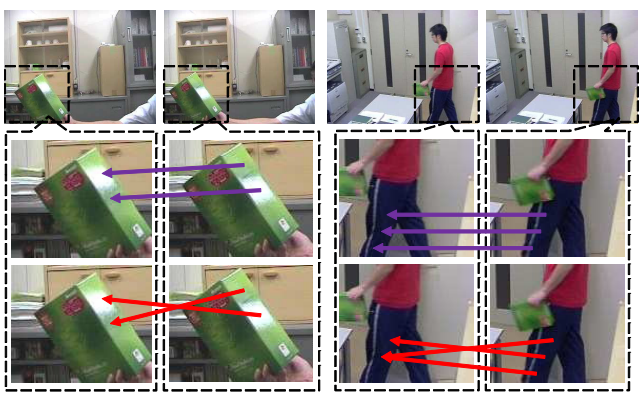
key words: *Motion estimation, tracking, structure matching, linear assignment*

1. Introduction

Moving object extraction and tracking from an image sequence captured by a static camera play important roles in various computer vision applications such as robot vision, simultaneous localization and mapping (SLAM), intelligent transport systems (ITS), and video surveillance systems. In order to achieve accurate action analysis of a moving object, it is necessary to obtain the corresponding relations between each part of an object in consecutive frames. Traditional tracking methods can be roughly classified into two categories: tracking method on object-level and that on pixel-level. For example, Mean-Shift [2] and Particle Filter [3], [4] track moving object on object-level and do not obtain the corresponding relation of each part in the object. On the other hand, optical flow methods [5], [6], SIFT method [7], and SURF [8] methods estimate pixel motion between two frames. These methods can obtain the corresponding relations on the level of pixel.

In a previous study, we developed a novel method, the exclusive block matching (EBM) method [1]. The EBM method realizes object tracking and detailed motion estimation which is robust against occlusions by solving the linear assignment problem that integrates object tracking, background subtraction, frame subtraction, and optimal matching between the current frame and multiple previous frames. This method performs quantization of flows on the pixel level to flows on the block level (block flow below) considering blocks show higher robustness than pixels. Although the EBM method computes coarse flows, the obtained flows that show the corresponding relations are very useful for action analysis.

In order to estimate motion and track moving objects, many studies have been carried out to determine a local feature descriptor for matching. In general, local feature descriptor is calculated using visual feature constraint which is defined as RGB data captured by camera or some values derived from RGB data such as intensity [9], HSV histogram [3], [4], edge histogram [10], and orientation histogram [11]. However, even robust local descriptors based on visual feature constraint are affected by several phenomena such as illumination change, viewpoint change, size change, and noise. These phenomena are very common for image sequences of real data. Object tracking and motion estimation becomes unstable because of these phenomena. Figure. 1 (a) and (b) show consecutive frames, respectively. In Fig. 1 (a), inaccurate matching result is caused by the situation that colors on the same region are totally different because of the reflected light. In Fig. 1 (b), Matching results are obtained randomly due to uniform color distribution. When human beings observe the two scenes in Fig. 1 (a) and (b), they can understand the correct correspondence relations right away. We believe that human beings make these



(a): Colors on the same region are totally different.

(b): Colors on different regions are almost same.

Fig. 1 An example to illustrate that visual feature matching is not sufficient for tracking. Blue arrow and red arrow mean accurate and inaccurate matching results, respectively.

Manuscript received July 13, 2012.
Manuscript revised November 25, 2012.

[†]The author is with the Hangzhou Dianzi University, Hangzhou, 310018 China.
[††]The authors are with the Tokyo University of Agriculture and Technology, Koganei-shi, 184–8588 Japan.

*The preliminary version was published by [1]

a) E-mail: lz1126@hdu.edu.cn
b) E-mail: ytomioka@cc.tuat.ac.jp
c) E-mail: kitazawa@cc.tuat.ac.jp

DOI: 10.1587/transinf.E96.D.1171

inferences based on the assumption that the structure of an object does not change abruptly. We refer to this assumption as the structure constraint. For tracking moving objects and estimating motion, not only the visual feature constraint but also the structure constraint is indispensable.

In this paper, we propose a method by considering the relative positions of blocks of an object in order to determine structure similarity and to perform structure matching. Similar to visual feature matching in the EBM method, we adopt linear assignment to perform structure matching. Considering both the visual feature constraint and the structure constraint simultaneously, the optimization problem can be expressed by the following equation:

$$E = \sum_{l} VS(l, f_l^k) + \lambda \sum_{l} \sum_{w \in N_l} SS(l, w, f_l^k, f_w^k) \qquad (1)$$

Here, l is a block in the current frame and w is a neighboring block of l. N_l is the neighboring area of l. f_l^k and f_w^k are blocks in previous frames, which are matched with l and w, respectively. VS is the visual feature similarity between blocks, and SS is structure similarity calculated by variation in the relative positions of neighboring blocks. λ is a weighting factor. The problem becomes a quadratic assignment problem which is NP-hard. The proposed method does not solve the problem of Eq. (1) directly; however, it formulates both visual feature matching and structure matching as linear assignment problems. Visual feature matching extracts blocks that belong to objects from the background. Structure matching of these blocks obtains the corresponding relations with minimum change of structure. Then the visual feature cost matrix is adjusted according to the results of structure matching in order to integrate visual feature matching and structure matching together.

Different from methods such as Particle Filter [3] and Mean-Shift [2], the proposed method aims to track moving object and obtain the corresponding relation of each part of object. In [12], a large displacement optical flow estimation method based on regions has been proposed. This method uses the orientation histogram and RGB color as descriptors of a region, and it avoids outliers by considering spatial smoothness. However, it is affected by the problem shown in Fig. 1. Some related studies [13], [14] have reported the use of the spatial constraint. These methods, based on graph matching between 2 sets of feature points, are only applicable to the situation wherein the structure of the graph of objects changes slightly. They cannot solve the problem shown in Fig. 1 because the graph of the moving object has changed. In contrast, the proposed method does not require graph structures, and it is applicable to such situations. A method based on assignment, called SoftAssign, has been proposed in [15]. This method obtains a matching between 2 point sets under an affine transformation. Different from this method, our proposed method focuses on both visual feature and structure. Another related study is shape tracking. For example, a previous method [16] uses level sets to track the contours of moving objects. Different from such methods that focus on the shape of the contour of a moving object, our method can obtain motion of each part of an object.

The remainder of this paper is organized as follows. In Sect. 2, we explain the outline of the EBM method proposed in [1]. In Sect. 3, we describe how to determine structure similarity, and we propose a method to integrate visual feature matching and structure matching together. Then, we present some experimental results in Sect. 4. Finally, in Sect. 5, we provide concluding remarks.

2. Exclusive Block Matching

In this section, we describe the original algorithm of the EBM method. In order to avoid the situation wherein the destinations of matched blocks are too close or overlap, the EBM method assumes that the structures of objects change smoothly. Under this assumption, blocks match in such a way that the destinations are mutually exclusive. Therefore, the optimal matching can be obtained using linear assignment. In this paper, we visualize the motion of a block as a flow and referred to as block flow.

2.1 Exclusive Block Matching Handling Multi-Frames

First, the input images are scanned by blocks to convert them into 1 dimensional data. Let the block size be $n \times n$ pixels and image size be $w \times h$ pixels. The number of blocks N is given by the equation $N = w/n \times h/n = W \times H$. Then, an $N \times N$ array consisting of the similarities (actually, difference measure or distance) is constructed between the blocks of current frame (Curr) and the blocks of the previous frame (Prev). Here, we adopt the Bhattacharyya distance [4] to define a distance on HSV histograms in order to measure the similarity between 2 blocks. Then, the basic matrix is expanded by adding the similarity matrix for $T - 1$ previous frames, the Bg array and the Create part. Figure 2 shows the final form of the matrix. In our method, the initial frame without any moving object is used as background image. We use IIR filter to update the background image. The elements of the Bg array are similarities between the blocks of the current frame and those of the background image. The Create array consists of a predetermined threshold value. The dots in the Bg and Create arrays mean only the diagonal elements of the Bg and Create arrays can be selected according to the reappearance of the background and creation of new blocks, respectively. Non-diagonal elements are filled with sufficiently large values.

The matrix dealing with multiple previous frames can discern certain situations such as occlusion and reappearance. For example, block A in the frame t-3 in Fig. 2 is matched with a block in the current frame. This block is regarded as occluded at t-1 and t-2. The matching problem can be mathematically expressed as follows:
Minimize

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

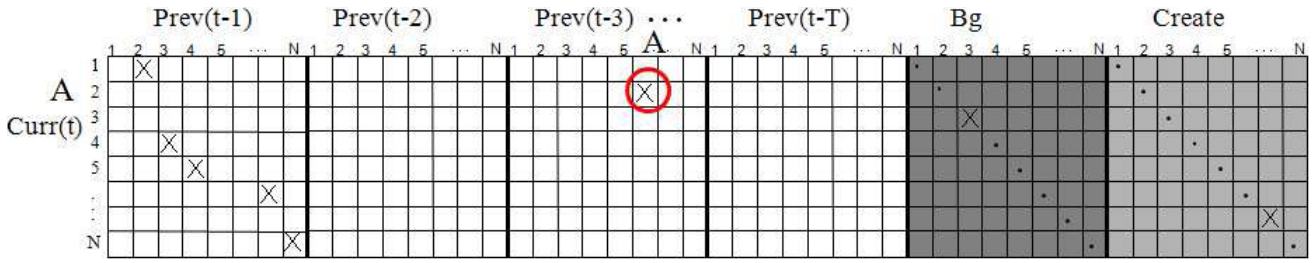


Fig. 2 Cost matrix dealing with multiple previous frames. The value in () denotes the time of the frame.

subject to

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

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

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

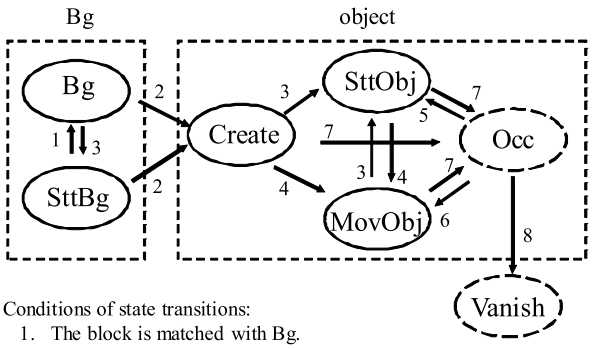
$$c_{ij} = \begin{cases} Dist\{Curr(t)_i, Prev(t-h)_j\} \\ \quad h = \{1, \ldots, T\}, \\ \quad i = \{1, \ldots, N\}, \\ \quad j = \{(h-1) \times N + 1, \ldots, h \times N\}, \\ Dist\{Curr(t)_i, Bg_j\} \\ \quad i = \{1, \ldots, N\}, \\ \quad j = \{T \times N + 1, \ldots, (T+1) \times N\}, \\ Threshold\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

In the above equations, T is the number of previous frames. $Dist\{Curr(t)_i, Prev(t-h)_j\}$ is the distance between the current frame(t)'s block i and the previous frame(t-h)'s block j. Here, the distance is the average value of the $Dist\{Curr(t)_i, Prev(t-h)_j\}$ and its 8-connected neighboring blocks (with the same object number). $Dist\{Curr(t)_i, Bg_j\}$ is the distance between the current frame's block i and the background's block j. $Threshold\ for\ creating$ is a fixed value selected when there is no block similar to it. If this value is selected, this block is regarded as creating of new block.

This is a type of linear assignment problem, and it can be solved by the Hungarian method [17].

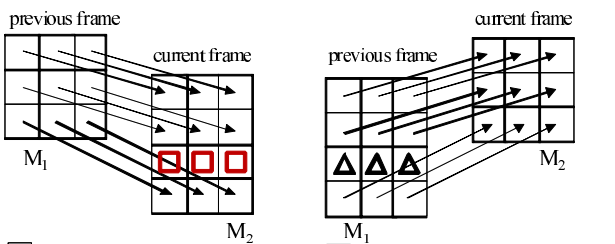
2.2 Restriction of Block State Transition

In the EBM method, a processing technique called restriction of block state transition is adopted to avoid a large amount of unnecessary calculations and to improve the stability of block matching. In the system, 7 kinds of block states are defined. They are Moving Object, Static Object, Create, Background, Static Background, Occlusion, and Vanish. The state of a block transits to another state obeying the restriction rules as shown in Fig. 3. In the cost matrix, we replace the cost with a very large value to avoid matching between unavailable state transitions. A detailed description of the EBM method can be found in our previous paper [1].



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. 3 Restriction of block state transition.



(a): An object scales up in a scene. (b): An object scales down in a scene.

Fig. 4 The EBM method can handle magnification and shrink by considering vanishing and creating.

When objects enter the scene, they appear as blocks with state Create. We adopt an 8 neighborhood component labeling algorithm to segment multiple objects and determine the object number of each Create block. As long as a block is matched with another block in the next frame, the states of blocks become Moving Object or Static Object and the object numbers are inherited.

2.3 Handling Magnification and Shrink on the Visual Feature Matching Stage

Although the EBM method assumes blocks match exclusively, it considering vanishing and creating is applicable for magnification and shrink on the visual feature matching stage. As shown in Fig 4 (a), we assume the size of a moving object changes from M_1 (in the previous frame) blocks to M_2 (in the current frame). If the object scales up in the current frame, only M_1 blocks are recognized as Moving Object or Static Object according to the processing of restriction of block state transition. Other $M_2 - M_1$ blocks are recognized as Create of new blocks in the current frame. Conversely, if the object scales down, M_2 blocks are recognized as Moving Object or Static Object in the current frame as shown in Fig. 4 (b).

3. Structure Matching for Shape Preservation

In this section, we describe structure matching and the method that integrates visual feature matching and structure matching together. In the proposed method, we assume the structures of objects do not change abruptly in two consecutive frames. Under this assumption, the relative positions of the blocks in each object are almost unchanged. Structure matching is to find the optimal matching that maps the blocks on an object in the previous frame onto the blocks on the corresponding object in the current frame with the minimum transformation cost. The transformation cost of each block is regarded as structure similarity. Such block correspondences can be obtained via linear assignment.

3.1 Transformation Cost

Now, we describe the calculation of transformation cost and method for structure matching. The transformation cost of each object is evaluated by the total migration length of blocks required for a transformation is calculated as follows. Let P be a set of previous frame's blocks included in the same object and matched with current frame's blocks through the visual feature matching stage. Let C be a set of the corresponding blocks in the current frame. The set C is superimposed on the set P so that the center of gravity of the set C coincides with that of the set P. The change

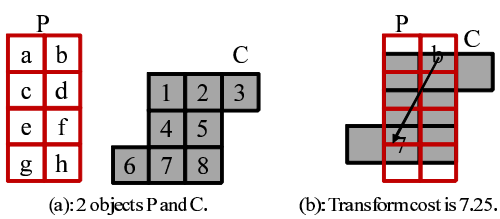
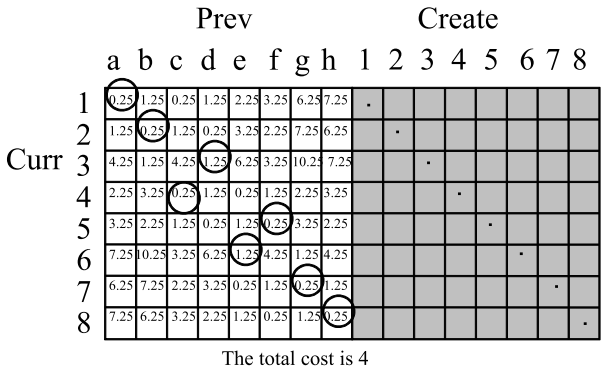
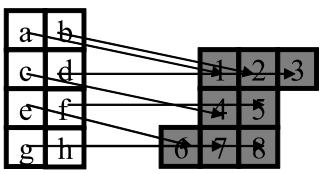


Fig. 5 P and C are the same object in the previous frame and the current frame. The change of relative positions of blocks are calculated by the square of the Euclidean distance. For example, If block 7 in C is matched with block b in P, the cost becomes $7.25=2.5^2 + 1^2$.



(a): The cost matrix for structure matching.



(b): Create block flows according to the result of structure matching.

Fig. 6 The cost matrix for structure matching. The circles represent the matching result with minimal total cost.

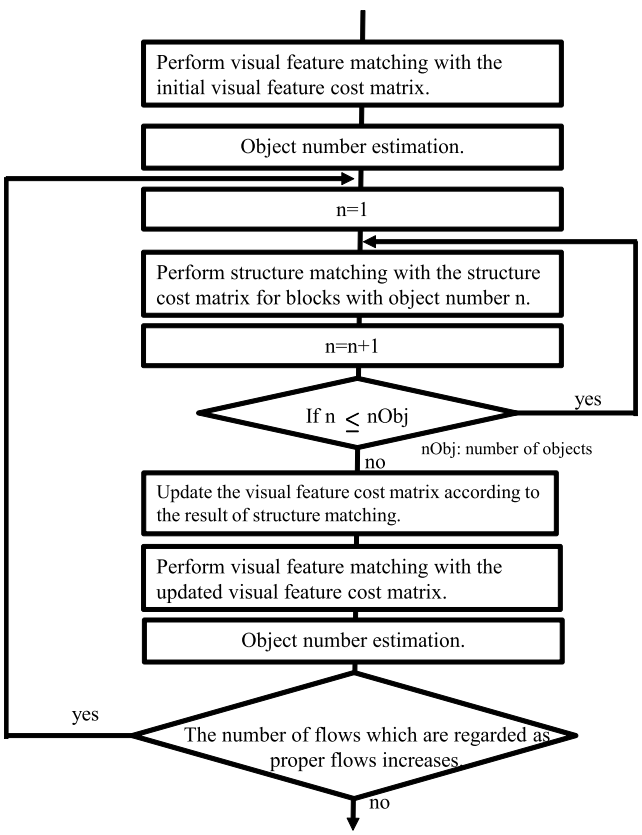


Fig. 7 Processing flow of proposed method.

of relative positions between a block in P and a block in C is calculated by an appropriate distance measure. Since we assume the structure of the object changes smoothly, block flows should be almost uniform. In this paper, the square of Euclidean distance is adopted as the distance measure. Then the distance is used as the transformation cost to construct a cost matrix. Figure 5 (b) shows an example of the square of the Euclidean distance between the objects shown in Fig. 5 (a). If block 7 in C is matched with block b in P, the cost becomes $7.25=2.5^2 + 1^2$.

Now, let the number of blocks of P and C be M (in this example, M is 8). The size of the cost matrix becomes $M \times M$. A Create array is also added to eliminate some matching whose cost is larger than a pre-specified threthold. Thus, the size of the cost matrix becomes $M \times 2M$, as shown in Fig. 6 (a). Similar to the visual feature cost matrix, non-diagonal elements are set as sufficiently large values in the Create array so that only diagonal elements can be selected. We create flows according to the corresponding relation obtained by the result of the assignment problem in Fig. 6 (a). The black circles in Fig. 6 (a) represent the matching result with minimal total cost of the structure matching. The created flows are shown in Fig. 6 (b).

The arithmetic expression for structure matching of one object can be written as follows:
Minimize

$$y = \sum_{i=1}^{M} \sum_{j=1}^{2\times M} r_{ij} s_{ij}, \tag{3}$$

subject to

$$\sum_{j=1}^{2\times M} r_{ij} = 1 \quad i = \{1, 2, \ldots, M\},$$

$$\sum_{i=1}^{M} r_{ij} \le 1 \quad j = \{1, 2, \ldots, 2 \times M\},$$

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

$$s_{ij} = \begin{cases} 1 \le j \le M \\ \quad Dist\{C_i, P_j\} \; for \; i = \{1, \ldots, M\}, \\ M + 1 \le j \le 2M \\ \quad Create \; TH \; for \; j = i, \\ \quad \text{sufficiently large value } for \; j \ne i, \end{cases}$$

where $Dist\{C_i, P_j\}$ is the distance between block i of C and block j of P, and $Create\ TH$ is a fixed value selected when there is no block in P similar to a block in C.

3.2 Iterative Approximate Method Considering Visual Feature and Structure Similarity

The matching result presented in the previous subsection is the optimal structure matching. From the viewpoint of visual feature, this is not the optimal solution. An ideal tracking should minimize both of visual feature and structure variations in Eq. (1). However, there is no effective method to solve visual feature matching and structure matching simultaneously within a short processing time. Therefore, we propose an approximate method that integrates visual feature matching and structure matching together as follows:

step 1 Perform visual feature matching with the initial visual feature cost matrix as described in Sect. 2 to extract blocks that are regarded as Moving Object or Static Object in the previous frames and current frame. However, the flows created by this step are ignored.

step 2 Perform structure matching between these extracted blocks, as described in Sect. 3.1, object by object, and create new block flows.

step 3 Reduce the costs of proper flows in the visual feature

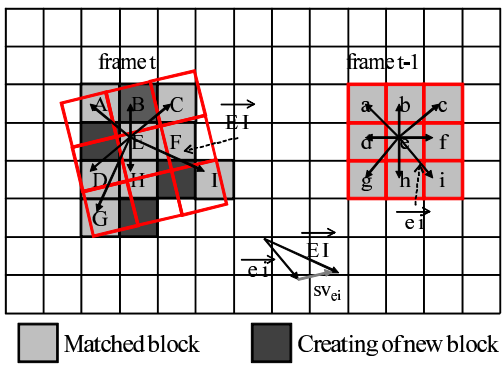


Fig. 8 Relative rotation direction of blocks.

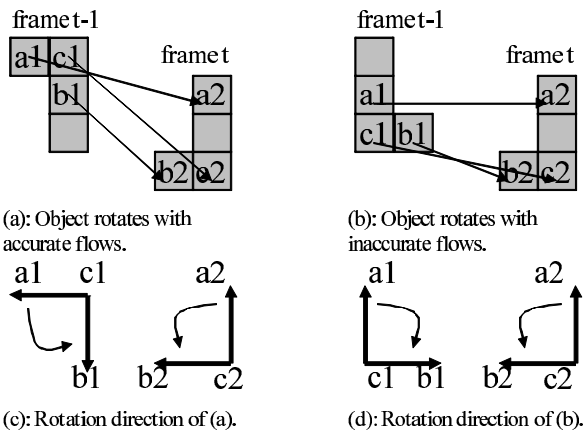


Fig. 9 Examples of accurate flow and inaccurate flow.

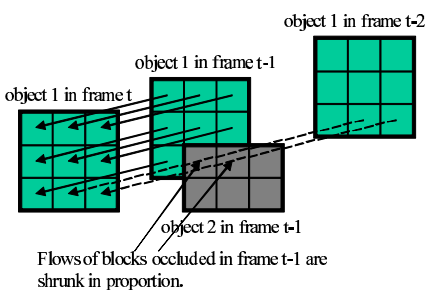


Fig. 10 Shrunk motion vectors of occluded blocks.

Table 1 Parameters of Experiments.

Image size	320×240 pixels	Block size	8×8 pixels
Number of blocks (1 frame)	40×30=1200	Number of previous frames	4
Size of visual feature cost matrix	1200×7200	Size of structure cost matrix	Depends on object size
PC for experiments	Core2 Duo3.00 GHz PC with 2 G RAM, WindowsXP		

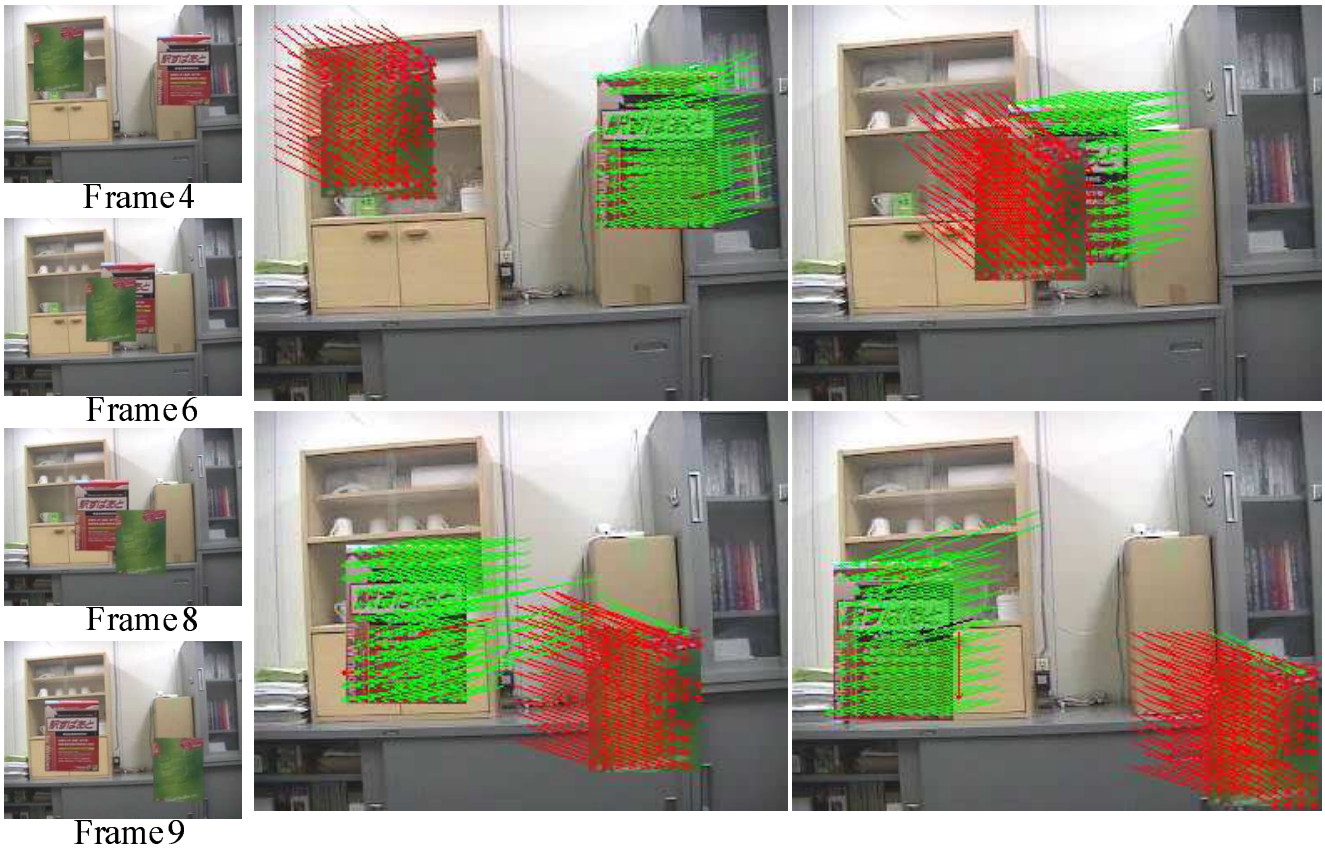


Fig. 11 Experimental results of 2 moving boxes obtained by the proposed method. We use different colors to distinguish flows of different object numbers. The description about how to determine the object number is at the end of the Sect. 2.2.

cost matrix to reflect the results of structure matching. If a block moves smoothly and if the visual feature difference between 2 frames is less than a threshold (the threshold is the same as the threshold of Create of visual feature cost matrix. It will be shown in Sect. 4.2.), the flow of this block is regarded as a proper flow. The smoothness is determined by comparing the block flow and the average flow of neighboring blocks. Note that only neighboring blocks with the same object number as the centric block are used for calculating the average flow. If the difference is less than a predetermined value (1 block in our experiments), this flow is regarded as a smooth flow. We reduce the cost value to 90% of the original value in the visual feature cost matrix according to the position of the flow. For example, we assume block 3 in the previous frame is matched with block 2 in the current frame. We reduce the cost in the second row of the third column.

step 4 Perform visual feature matching with the updated visual feature cost matrix.

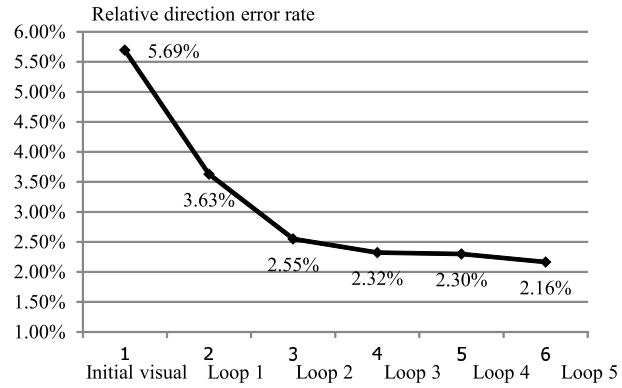
step 5 Repeat step2, step 3, and step 4 while the number of flows which are regarded as proper flows increases.

Figure 7 shows the processing flow chart of the proposed method.

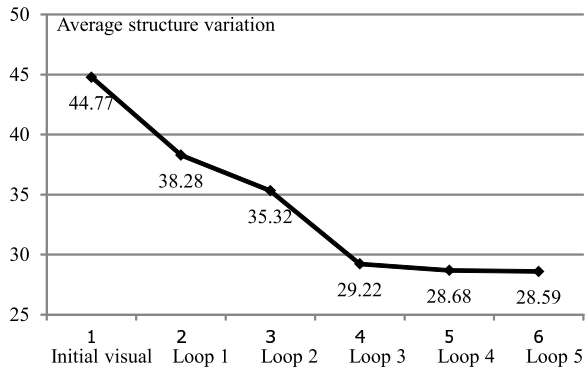
4. Experimental Results

4.1 Evaluation of Error Rate

The proposed method can handle many types of object motion such as rotation, magnification, occlusion, and reappearance. It is necessary to define some criterions to measure the accuracy of flows. We assume that the appearance of moving objects change smoothly. For each block, the relative directions of its neighboring blocks in this object are almost kept. As shown in Fig. 8, an object moves with motions of rotation and magnification. Blocks $a \sim i$ of the



(a): Relative direction error rate.



(b): Average structure variation.

Fig. 12 Relative direction error rate and average structure variation of experimental results of 2 moving boxes.

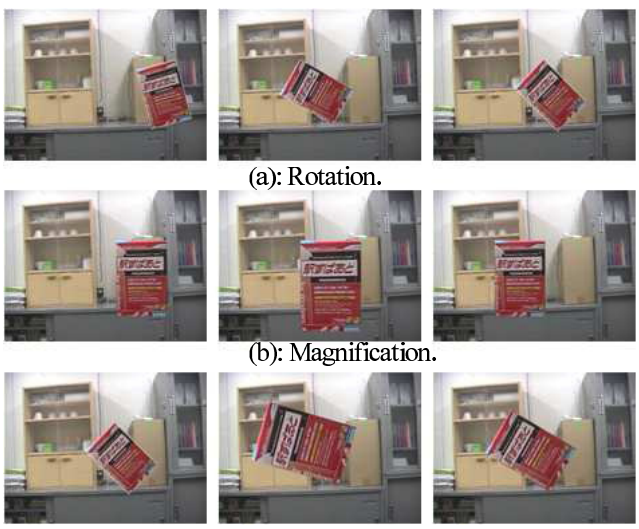


Fig. 13 CG Data with motion of rotation and magnification.

Table 2 Results of CG data with motion of rotation and magnification. EBM represents the result of original EBM method. PM represents the that of the proposed method. In the table TNP and NBO mean the average total number of pairs and the average number of blocks in objects of each frame, respectively.

	rder EBM	rder PM	asv EBM	asv PM
rot	4.68%	1.99%	12.5	4.7
mag	2.07%	0.12%	11.2	7.9
rotmag	6.34%	3.20%	26.4	14.4
	TNP EBM	**TNP PM**	**NBO EBM**	**NBO PM**
rot	665	665	98	98
mag	762	764	117	117
rotmag	741	741	111	111

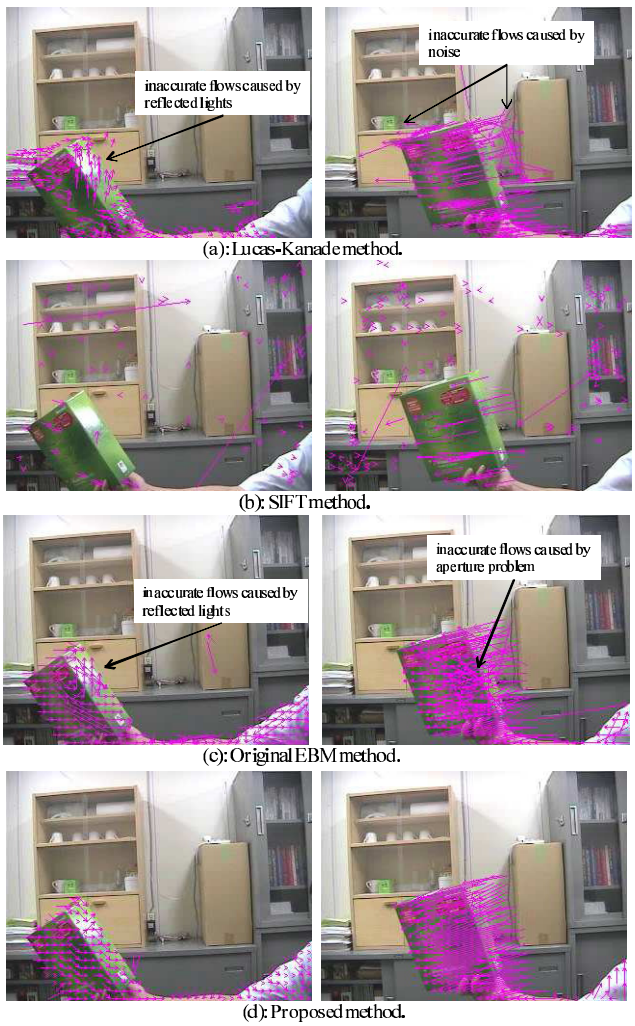


Fig. 14 Comparison with optical flow method and SIFT method.

object in frame $t-1$ are matched with blocks $A \sim I$ in frame t, respectively. For a centric block (e.g., block e), the relative directions of its neighboring blocks are invariant. We focus on this point, and we define the first evaluation method called relative direction error rate (rder below) considering the motion vectors of 8-connected neighboring blocks of each block in the object. Figure 9 shows 2 moving objects rotating in the scene. Although the object in Fig. 9 (a) is rotating, the relative directions of block $a1$ and block $b1$ observed from block $c1$ do not change. We regard these flows as accurate flows. On the other hand, the relative

Fig. 15 Results of real data with several actions obtained by the proposed method.

direction in Fig. 9 (b) changes, and these flows are regarded as inaccurate. We regard each block in the object as a centric block in turn, and we check each pair of blocks around it. Here only the pairs of blocks that belong to the same object as the centric block are checked. The relative direction error rate is defined by Eq. (4).

$$rder = \frac{Number\ of\ inaccurate\ pairs}{Total\ number\ of\ pairs} \tag{4}$$

The second criterion is called average structure variation (asv below). In our previous study [18], we adopted norm of difference of two vectors to calculate the variations of structure and introduce it into the optimization problem as structure constraint. We assume that blocks Q_i and Q_j in the previous frame are matched with the current frame's blocks P_i and P_j. The vectors $\overrightarrow{P_iP_j}$ and $\overrightarrow{Q_iQ_j}$ represent relative positions. Then, the variations in the relative positions are calculated from the norm of the difference of two vectors as follows:

$$sv_{ij} = \|\overrightarrow{P_iP_j} - \overrightarrow{Q_iQ_j}\|. \tag{5}$$

For example, the variation of relative position of block e and i are sv_{ei} in Fig. 8. Then the average structure variation is calculated by the following equation:

$$asv_n = \frac{\sum_{i=1}^{M}\sum_{j=1}^{M} sv_{ij}}{M^2}. \tag{6}$$

$$asv = \frac{\sum_{i=1}^{N} asv_n}{N}. \tag{7}$$

Here, M is the number of blocks in an object and N is the number of objects.

When we calculate the rder and asv, if blocks are occluded in the previous frame, we shrink their flows proportionally as shown in Fig. 10.

4.2 Parameters of Experiments

Table 1 lists some parameters of our experiments. We adopted the Munkres assignment algorithm [17], which is an implementation of the Hungarian method, to solve the linear assignment problem. The cost matrix is very sparse, even though its size is large. Therefore, the processing speed is improved by sparse-matrix calculation. The processing time for visual feature matching is approximately 0.5 $s \sim 2$ s. It depends on the number of blocks that belong to objects in the scene. The processing time for structure matching is less than 0.1 s. We normalize the Bhattacharyya distance in the range of 0 to 1000. The threshold value of Create is set to 400. The threshold value of Create of the structure cost matrix for structure matching is set to 40. We determine both threshold values of visual feature cost matrix and structure cost matrix through experiments on CG data shown in Fig. 11. Then the parameters are fixed for all experiments.

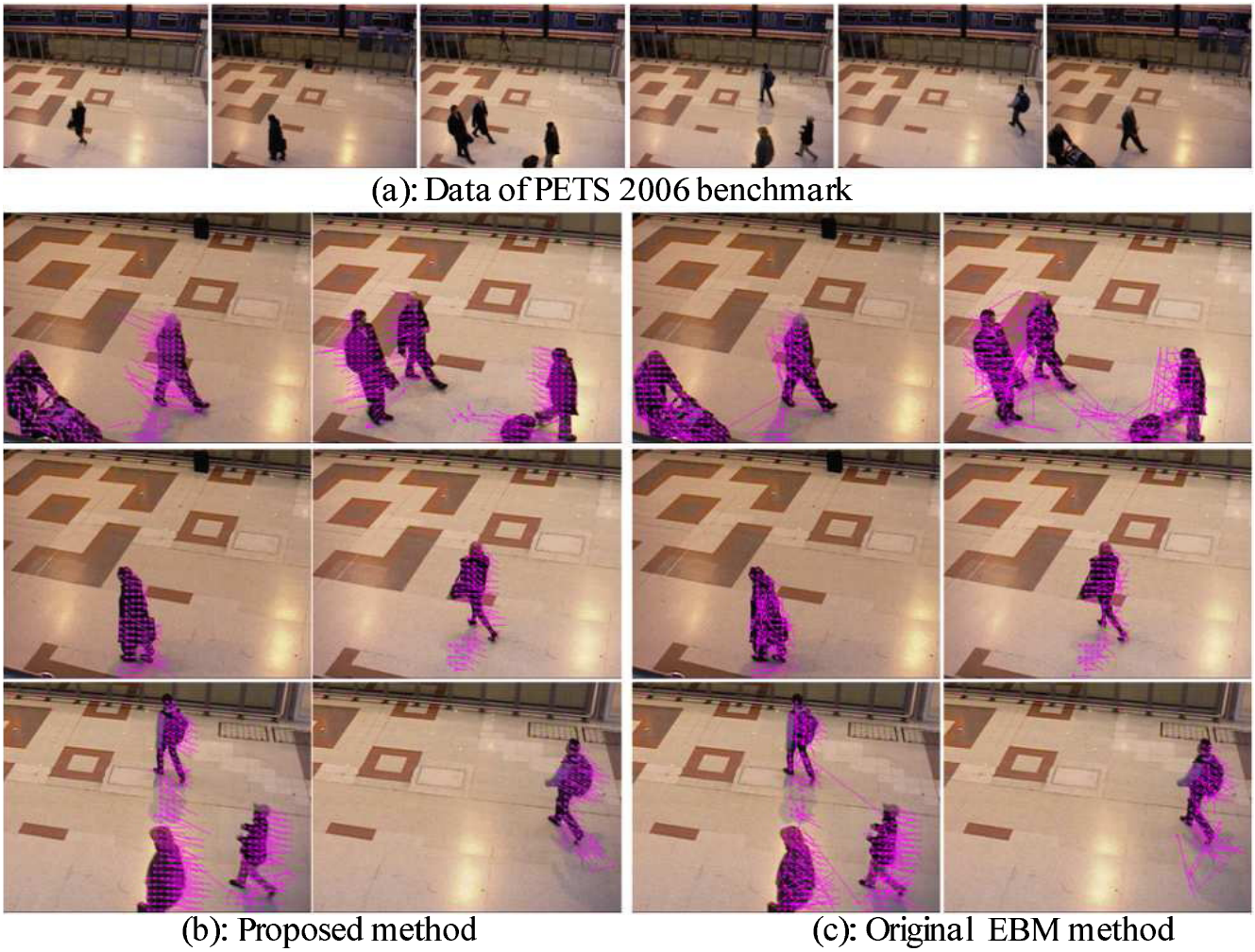


Fig. 16 Results of PETS 2006 benchmark data obtained by the proposed method.

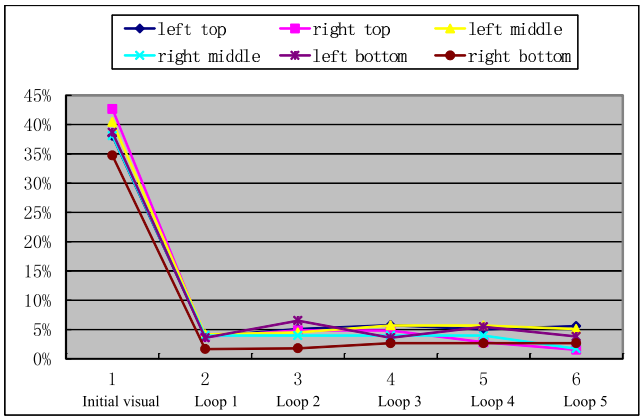
According to our experimental results, extractions in different scenes have little dependence on the threshold values of visual feature cost matrix and structure cost matrix. When the threshold value of visual feature cost matrix is changed in the range of 300 ∼ 500, experimental results are almost the same. Also, when the threshold value of structure cost matrix is changed in the range of 20 ∼ 50, experimental results are almost the same. The large values in Eqs. (2) and (3) must be sufficiently larger than both cost values. We used 99999 in our experiment.

4.3 CG Data of Two Moving Boxes

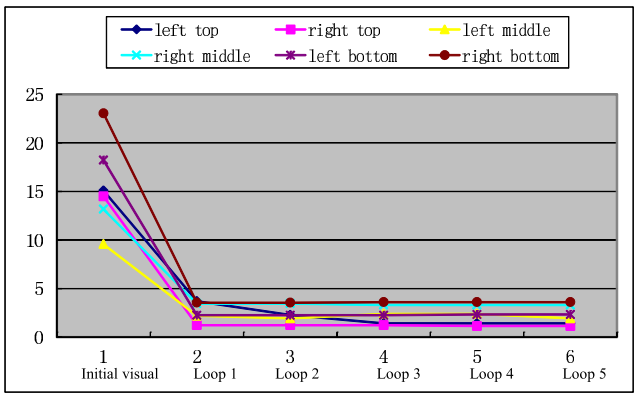
First, we present the experimental results of CG generated data. Two boxes move parallelly and occlusion occurs at frames 6 and 7. Figure 11 shows the obtained flows. We use different colors to distinguish flows of different object number. In Fig. 11, a few flows are mismatched to blocks of another object. This is because their colors are very similar. These flows are improper, and they are counted up for the errors. Figure 12 (a) and (b) show the relative direction error rate and average structure variation of each frame, respectively. According to the result, the proposed method has reduced the error rate significantly in the first three loops. From the 3rd loop, the error rate tends to the same level. Another example of rder and asv for PETS2006 data will be shown in Sect. 4.6. The error rates are reduced significantly in the first two loops. Although the speed of convergence depends on the size of moving objects, the error rate tends to decrease rapidly in the first few loops. In the experiments, we only repeat the iteration for 2 times considering efficiency.

4.4 CG Data with Motion of Rotation and Magnification

Then we give the experimental results of 3 groups of CG Data with motion of rotation and magnification as shown in Fig 13. Table 2 give the results of these 3 groups data. We also give the average total number of pairs (TNP in the figure) and the average number of blocks in objects (NBO in the figure) of each frame. According to the results, we can see that two measures are reduced significantly.



(a): Relative direction error rate.



(b): Average structure variation.

Fig. 17 Rder and asv of results of PETS 2006 benchmark data. Each line plot represents Rder/asv for a frame in Fig. 16 (b).

4.5 Comparison with Other Methods

We compare the proposed method with the traditional tracking methods which can obtain the corresponding relation in detail. They are Lucas-Kanade method, SIFT feature point matching and EBM method without structure matching. The results of the Lucas-Kanade method are calculated using the OpenCV library [19] and the code of SIFT method is from Rob Hess's homepage [20]. We adjust the parameters to obtain the best possible results. The experimental results shown in Fig. 14 (a) indicate many inaccurate flows caused by reflected light and noise in the result of the Lucas-Kanade method. In the result of the SIFT method, the extracted feature points are insufficient for tracking the box. The original EBM method obtains dense block flows. However, the flows are also affected by reflected light, noise and aperture problem. The proposed method avoids this problem, and the obtained dense block flows are nearly accurate.

4.6 Real Data

Figure 15 (a) shows the results of a person performing different actions. Figure 16 (a) are PETS 2006 benchmark data [21]. Figure 16 (b) are parts of magnified results. In these results, our method not only tracks the moving objects but also obtains the motion of each part. This is useful for motion analysis. Figure 15 (b) and Fig. 16 (c) show the results obtained by the original EBM method. Colors of blocks on the cloth of people are similar. Matching on these areas are very unstable because of the aperture problem. The proposed method considering structure similarity improves this problem significantly. In Fig. 17 (a) and (b), we also give the rder and asv of the frames in Fig. 16 (b). However, some inaccurate flows still remain. The reason is that the motions of different parts (e.g., hand and body) of one object are independent. Another problem is that some inaccurate extraction caused by shadow.

5. Conclusion

In this paper, we proposed a new method that integrates visual feature matching and structure matching together. As compared to matching based only on visual feature, the proposed method reduced the error rate significantly. Moreover, both visual feature matching and structure matching are formulated as linear assignment problem. This enables us to conveniently realize our method via hardware implementation. In the future, we plan to further enhance this method by considering motion segmentation and shadow detection.

Acknowledgements

This research was partially supported by the Ministry of Education, Culture, Sports, Science and Technology (MEXT), Grant-in-Aid for Scientific Research (C), 22500149. 2010.

References

[1] Z. Li, K. Yabuta, and H. Kitazawa, "Exclusive block matching for moving object extraction and tracking," IEICE Trans. Inf. & Syst., vol.E93-D, no.5, pp.1263–1271, May 2010.
[2] D. Comaniciu, P. Meer, and S. Member, "Mean shift: A robust approach toward feature space analysis," IEEE Trans. Pattern Anal. Mach. Intell., vol.24, pp.603–619, 2002.
[3] P. Pe'rez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," European Conference on Computer Vision, pp.661–675, 2002.
[4] K. Nummiaro, E. Koller-meier, and L.V. Gool, "A color-based particle filter," Image Vis. Comput., pp.53–60, 2002.
[5] B.K.P. Horn and B.G. Schunck, "Determining optical flow," Artif. Intell., vol.17, pp.185–203, 1981.
[6] J. yves Bouguet, "Pyramidal implementation of the lucas kanade feature tracker," Intel Corporation, Microprocessor Research Labs, 2000.
[7] D. Lowe, "Object recognition from local scale-invariant features," Proc. Int. Conf. Computer Vision 2, pp.1150–1157, 1999.
[8] H. Bay, T. Tuytelaars, and L.V. Gool, "Surf: Speeded up robust features," European Conference on Computer Vision, pp.404–417, 2006.
[9] Stauffer and W.E.L. Grimson, "Adaptive background mixture models for real-time tracking," IEEE Int. Conf. Computer Vision and Pattern Recognition, 1999.
[10] K. She, G. Bebis, H. Gu, and R. Miller, "Vehicle tracking using online fusion of color and shape features," IEEE Int. Conf. Intelligent Transportation Systems, pp.731–736, 2004.

[11] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," IEEE Int. Conf. Computer Vision and Pattern Recognition, pp.886–893, 2005.
[12] T. Brox, C. Bregler, and J. Malik, "Large displacement optical flow," IEEE Int. Conf. Computer Vision and Pattern Recognition, pp.41–48, 2009.
[13] T. Caetano, J. McAuley, L. Cheng, Q. Le, and A. Smola, "Learning graph matching," IEEE Trans. Pattern Anal. Mach. Intell., vol.31, pp.1048–1058, 2009.
[14] H. Jiang, S.X. Yu, and D.R. Martin, "Linear scale and rotation invariant matching," IEEE Trans. Pattern Anal. Mach. Intell., vol.33, pp.1339–1355, 2011.
[15] S. Gold, A. Rangarajan, C. ping Lu, and E. Mjolsness, "New algorithms for 2d and 3d point matching: Pose estimation and correspondence," Pattern Recognit., vol.31, pp.957–964, 1997.
[16] A. Yilmaz, X. Li, and M. Shah, "Contour based object tracking with occlusion handling in video acquired using mobile cameras," IEEE Trans. Pattern Anal. Mach. Intell., vol.26, no.11, pp.1531–1536, 2004.
[17] "Munkres assignment algorithm," http://csclab.murraystate.edu/bob.pilgrim/445/munkres.html
[18] Z. Li, K. Tomotsune, Y. Tomioka, and H. Kitazawa, "Template matching method based on visual feature constraint and structure constraint," IEICE Trans. Inf. & Syst., vol.E95-D, no.8, pp.2105–2115, Aug. 2012.
[19] "Opencv," http://opencv.jp/document/opencvrefcv.html
[20] "Code of sift keypoint detector," http://web.engr.oregonstate.edu/~hess/index.html
[21] "Code of sift keypoint detector," http://www.cvg.rdg.ac.uk/PETS2006/data.html

Zhu Li received the B.S degree in electrical and information engineering from ZheJiang University of Technology, China, in 2005. Then, he received the M.S., and D.E., degree in electrical and electronic engineering from Tokyo University of Agriculture and Technology, Japan, in 2009, and 2012. Since 2012, he has been a lecturer at Hangzhou Dianzi University. His research interests are image processing and object detection.

Yoichi Tomioka received his B.E., M.E., and D.E., degrees from Tokyo Institute of Technology, Tokyo, Japan, in 2005, 2006, and 2009, respectively. He was with the Tokyo Institute of Technology as a research associate in 2009. Currently, he is with the Division of Advanced Electrical and Electronics Engineering, Tokyo University of Agriculture and Technology, as an assistant professor, a post he has held since 2009. His research interests are image processing, VLSI package design automation, and combinational algorithms. He is a member of IEEE and IPSJ.

Hitoshi Kiatazawa 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 has been a professor at Tokyo University of Agriculture and Technology. His research interests are VLSI CAD algorithms, computer graphics and image processing. He is a member of IPSJ and IEEE.