Article Information

Title	Template Matching Method Based on Visual Feature Constraint and Structure Constraint
Authors	Zhu LI, Kojiro TOMOTSUNE, Yoichi TOMIOKA, and Hitoshi KITAZAWA
Citation	IEICE TRANSACTIONS on Information and Systems, Vol.E95-D, No.8, pp.2105-2115
Copyright	
IEICE Transactions Online URL	https://search.ieice.org/

PAPER

Template Matching Method Based on Visual Feature Constraint and Structure Constraint

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

SUMMARY Template matching for image sequences captured with a moving camera is very important for several applications such as Robot Vision, SLAM, ITS, and video surveillance systems. However, it is difficult to realize accurate template matching using only visual feature information such as HSV histograms, edge histograms, HOG histograms, and SIFT features, because it is affected by several phenomena such as illumination change, viewpoint change, size change, and noise. In order to realize robust tracking, structure information such as the relative position of each part of the object should be considered. In this paper, we propose a method that considers both visual feature information and structure information. Experiments show that the proposed method realizes robust tracking and determine the relationships between object parts in the scenes and those in the template.

key words: *template matching, visual feature constraint, structure constraint, linear assignment, genetic algorithm*

1. Introduction

Template matching for image sequences captured with a moving camera plays an important role for motion picture analysis. Template matching is required for many applications such as Robot Vision, simultaneous localization and mapping (SLAM), intelligent transport systems (ITS), and video surveillance systems. Moreover, it is useful for the motion analysis of moving objects if the relation between the parts in the template image and the corresponding parts of the object in the current frame can be obtained.

Many algorithms that employ visual feature descriptors such as HSV, HOG, and SIFT have been proposed for template matching applications. For example, Mean-Shift [1] and Particle Filter [2], [3] are very popular in current research. In general, Mean-Shift calculates the similarities between templates and objects by distance measure on HSV histograms. In the algorithm of Particle Filter, the weight of each particle describes its likelihood and the weights of all particles represents the estimate of the posterior. The weight of each particle can be computed using many methods, one of them is template matching which normally calculates the similarities by using HSV histograms. However, matching becomes unstable when the visual features of the object in the current frame are affected by changes in its environment, which is very common for image sequences of real outdoor data. As shown in Fig. 1, template matching in such a situation poses several challenges such as illumination change, viewpoint change, size change, and noise. These phenomena lead to inaccurate template matching because they considerably alter the visual features of the object in the current frame from those of the template.

In order to resolve such issues, several studies have been conducted with the aim of determining a feature descriptor that is robust to changes in the environment. The SIFT [4] feature is robust to many conditions and can obtain the relation between each part of an object and the corresponding part of the template. However, in general, certain parts in the image do not contain discernible feature points. Moreover, the descriptor is not always stable when the visual features of the object in the current frame change.

Even though the visual feature information of certain parts of an object is completely different from that of the parts of a template, the similarity between the part of the object and the corresponding part of the template can be clear to the human eye. We believe that human beings make this inference based on the assumption that the structure of an object does not change radically. We refer to this assumption as the structure constraint. On the other hand, the

Fig. 1 Images on the right are templates. Images on the left are current frames that include the object with illumination change, viewpoint change, size change, and noise.

Manuscript received October 14, 2011.
Manuscript revised February 2, 2012.

[†]The authors are with the Tokyo University of Agriculture and Technology, Fuchu-shi, 183–8538 Japan.
a) E-mail: lizhu@m.ieice.org
b) E-mail: ktomotsune@live.jp
c) E-mail: ytomioka@cc.tuat.ac.jp
d) E-mail: kitazawa@cc.tuat.ac.jp

DOI: 10.1587/transinf.E95.D.2105

constraint derived from camera input in the form of similarities obtained using HSV, HOG, and SIFT is referred to as the visual feature constraint in this paper. For tracking and matching a moving object, both visual feature and structure constraints are indispensable.

In this paper, we propose a method that employs both visual feature and structure constraints. This method formulates the template matching problem as a block matching problem. Visual constraint is expressed in terms of the distances between the visual similarities of the blocks. Structure constraint is expressed by the total absolute change of distance between blocks in the same object. The matching problem is a quadratic assignment problem that is NP-hard. Presently, there is no effective method to solve this problem within a short processing time. Therefore, we adopt genetic algorithm [5] to search the approximate solution.

Some related studies have reported the use of the spatial constraint. The traditional optical flow method [6] that uses smoothness term is not applicable to frames where object motions are very large. In [7], 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. It is affected by environmental change. Graph structure based methods [8], [9], which perform graph matching between 2 sets of feature points, are only applicable to the situations wherein the structure of the graph of objects changes slightly. These methods are not applicable if the graph structure of the target object in the scenes cannot be created. This problem is very common because stable feature points cannot always be extracted from the target object in scenes. On the other hand, our method, which does not require the graph structures of the moving object exhibits higher robustness. Another related study is shape tracking. For example, the method in [10] and [11] use level sets to track contour of moving objects. The method [12] calculates similarity by the combination of four features which are HSV, vertical edge, horizontal edge, and diagonal edge. Different from these methods, our method that focus on the structures of objects can obtain motion of each part of an object.

The remainder of this paper is organized as follows. In Sect. 2, we explain the block matching process with consideration of the visual feature constraint. In Sect. 3, we describe our method for determining structure similarity. In Sect. 4, we describe how to solve the optimization problem by adopting genetic algorithm. The results of experiments are presented in Sect. 5. Real-time tracking requires a large amount of computational resources. We explain that our method is suitable for parallel processing in Sect. 5. Finally, in Sect. 6, we provide concluding remarks.

2. Exclusive Block Matching for Template Matching

In this section, we describe the exclusive block matching method for template matching considering the visual feature constraint.

2.1 Basic Cost Matrix for Template Matching Considering Visual Feature Constraint

In this subsection, we explain the method for creating the basic cost matrix for template matching which is similar to our previous study in [13]. We assume block matches in such a way that matched blocks in the current frame are mutually exclusive. First, we scan the current frame and template image by block to convert the images into one-dimensional data. If we assume that the block size is $n \times n$ pixels and that the width and height of the current frame are w and h, respectively, then the number of blocks, N, is given by the equation $N = w/n \times h/n$; shown in Fig. 2. Next, we scan the template blocks in the same way. The parts that belong to the background are eliminated by a mask. If more than half the pixels of a block are background pixels, this block is eliminated. Let us suppose that the number of blocks in the template is M. We build an $M \times N$ array that consists of visual feature similarities (actually difference measure or distance [2]) between the current frame's blocks (Curr Blk) and the template's blocks (Temp Blk). The matrix is shown in Fig. 3. Next, one-to-one matching between Curr Blk and Temp Blk should be performed in such a way that the total distance is minimized.

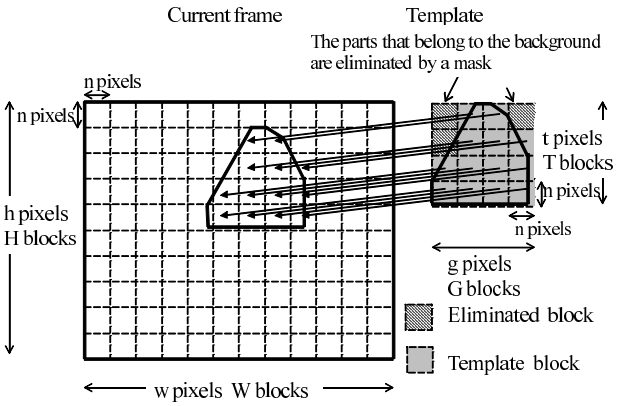

Fig. 2 Scanning of the current frame and template into one-dimensional data.

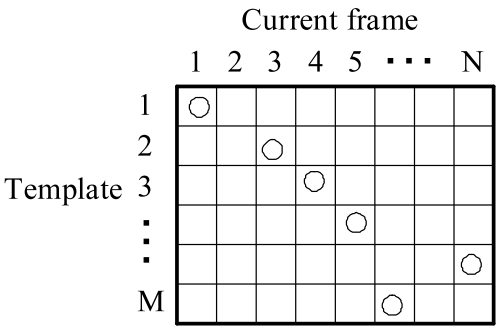

Fig. 3 Basic cost matrix for template matching.

2.2 Expansion of Basic Cost Matrix Considering Vanishment and Scale Variant

When the scale changes or target object is occluded by other objects, it is impossible to realize one-to-one matching. In order to solve these problems, we expand the basic cost matrix.

If the object is scaled up in the current frame, blocks of the template are matched with the most similar blocks of the object in the current frame, as shown in Fig. 4 (a). When the object is scaled down, the presence of similar blocks in the current frame results in inaccurate matching, as shown in Fig. 4 (b). In order to solve this problem, we expand the cost matrix. When the object is scaled down in the scene, the number of blocks in the object image decreases from M to M_{min}. The multiple K is calculated by the equation $K = \left[\frac{M}{M_{min}}\right]$. In practical experiments, we set the value of K to 3, which means that the number of blocks in the object image can decrease from M to $\frac{M}{3}$. Figure 5 illustrates an example where $K = 3$. The array of the current frame is expanded to $3 \times N$ by adding 2 current frames. The elements of the Current frame 2 and Current frame 3 arrays are the same as those of Current frame 1 array. Next, we add a Vanish array that consists of a predetermined threshold value, as shown in Fig. 5. In the Vanish array, only the diagonal elements can be selected. If a block is matched with this part of the cost matrix, it is regarded as a vanishing block. The final size of the cost matrix becomes $M \times (3 \times N + M)$. The arithmetic expression is written down as follows:
Minimize

$$C = \sum_{i=1}^{M} \sum_{j=1}^{K \times N + M} p_{ij} c_{ij}, \tag{1}$$

subject to

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

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

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

$$c_{ij} = \begin{cases} dist\{Template_i, Current_j\} \\ \quad i = \{1, \ldots, M\}, \\ \quad j = \{1, \ldots, K \times N\}, \\ threshold\ for\ vanishing \\ \quad i = \{1, \ldots, M\}, \\ \quad j = \{K \times N + 1, \ldots, K \times N + M\}. \end{cases}$$

K: $K = \left[\frac{M}{M_{min}}\right]$. M is the number of blocks in the template, and M_{min} is the assumed minimum number of blocks in the target object.

$dist\{Template_i, Current_j\}$: Distance between block i of the template and block j of the current frame.

threshold for vanishing: A predetermined threshold value. If this value is matched, this block is regarded as vanishing.

This is a type of linear assignment problem and can be solved by the Hungarian method [14].

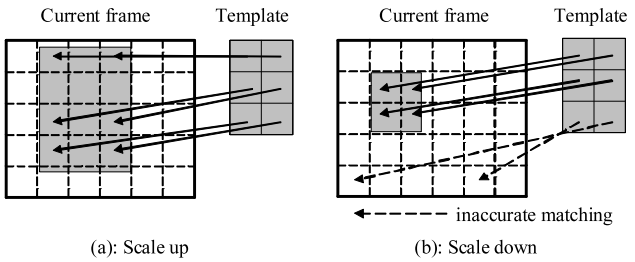

Fig. 4 Examples of Scale Variant. Blocks of the template are matched with the most similar blocks of the object in the current frame when the object is scaled up as shown in (a). Scaling down the object as shown in (b), results in inaccurate matching.

2.3 Calculation of Similarity

2.3.1 Visual Feature Similarity

In our study, the visual feature similarity between 2 blocks is calculated by the combination of three measures. It is calculated as follows:

$$c_{ij} = \alpha D_{HSV} + \beta D_{HOG} + (1 - \alpha - \beta) D_{HOG\ Context}, \tag{2}$$

c_{ij} represents the similarity between block i in the template and block j in the current frame. D_{HSV} and D_{HOG} are, respectively, the distances between the HSV histograms [2] of

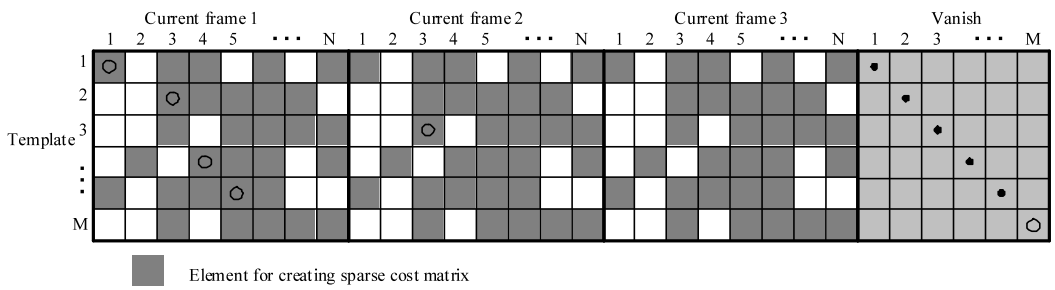

Fig. 5 Expansion of cost matrix for template matching.

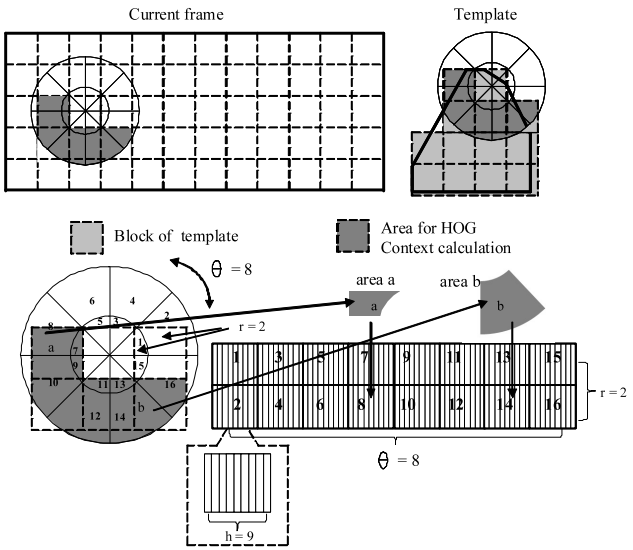

Fig. 6 Example to show the calculation of HOG Context histogram.

two blocks and HOG histograms [15] of two blocks. We also introduce the HOG Context histogram described in the next subsection in order to reflect the oriented gradient information around the blocks. This is an enhancement of the GLOH method [16]. In order to improve the processing speed, we use the HOG feature instead of the SIFT feature. In Eq. (2), $D_{HOG\ Context}$ is the distance between two HOG Context histograms, those of block i and block j. α and β are weighting factors. In this paper, all distances are calculated using the Bhattacharyya distance.

2.3.2 HOG Context Histogram

Here, we describe the approach to calculating the HOG Context histogram for template matching. HOG features and HOG context histograms are different in two points.

1. HOG context histograms reflect the oriented gradient information around blocks.
2. HOG context histograms reflect relative positions of HOG features by using shape context descriptors.

For a block of the template, the area around it is divided into $r \times \theta$ sub-areas, as shown in Fig. 6 [17]. We calculate the gradient and orientation of each pixel. The orientation is divided into h directions. The number of sub-areas in Fig. 6 is 16 ($r = 2$, $\theta = 8$) and the number of directions h is 9. Therefore, the number of bins of the HOG Context histogram becomes 144 (16×9). If the position of a pixel within a sub-area is inside the template, the pixel on this position is considered in the calculation of the HOG Context histogram. Similarly, the pixel on the same relative position in the current frame is calculated for the HOG Context histogram. Figure 6 illustrates two examples. For sub-areas 8 and 14, only the pixels within area a and b are available to calculate the HOG Context histogram. Next, each available pixel casts a weighted vote for an orientation-based histogram bin based on the values of the gradient.

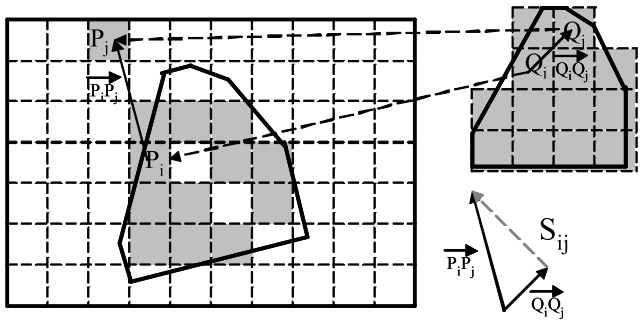

Fig. 7 Variations in the relative positions are calculated from the norm of the difference of two vectors.

3. Block Matching based on Both Visual Feature Constraint and Structure Constraint

As described in the introduction, the structure constraint, which assumes that the relative position of blocks does not change abruptly, is indispensable for template matching. Here, we will explain the formulation of the structure constraint.

The relative positions of the blocks can be expressed using vectors. Therefore, we adopted norm of difference of two vectors to calculate the variations in the relative positions of blocks. In Fig. 7, template's blocks Q_i and Q_j are matched with the current frame's blocks P_i and P_j, respectively. 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:

$$S_{ij} = \|\overrightarrow{P_iP_j} - \overrightarrow{Q_iQ_j}\|. \tag{3}$$

Next, the total structure variation is calculated by the following equation:

$$S = \sum_{i=1}^{M} \sum_{j=1}^{M} S_{ij}. \tag{4}$$

Considering both the visual feature and structure constraints simultaneously, the optimization problem can be expressed by the following equation:

$$E = w * C + S, \tag{5}$$

where C is the total cost of visual feature distances in Eq. (1), S is the total structure variation in Eq. (4), and w is a weighting factor.

4. Solution of Optimization Problem by Adopting Genetic Algorithm

The optimization of E in Eq. (5) becomes a quadratic assignment problem that is NP-hard. As there is no effective method for solving this problem within a short processing time, we adopt genetic algorithm (GA) to search for the approximate solution in this paper.

4.1 Genetic Algorithm for Block Matching

In this subsection, we explain the outline of the GA. First, we create a square cost matrix with dummy values. The real size of the cost matrix is $(K \times N + M) \times (K \times N + M)$. For convenience in explanation, we assume that $(K \times N + M) = 8$ and provide an example of an 8×8 cost matrix, as shown in Fig. 8 (a). Two rows are filled with dummy values. Next, a candidate solution is encoded to create individuals, as shown in Fig. 8 (b). The next step involves generation of new individuals. As shown in Fig. 9, we employ order crossover (OX) [18] for crossover processing and exchange mutation (EM) for mutation processing. The fitness of an individual is calculated according to Eq. (5). Here, the part of the cost matrix with dummy values is not employed in the calculation of fittness. Tournament selection is then employed to select individuals with the best fitness to form a new population. The generation process is repeated until a fixed number of generations has been reached.

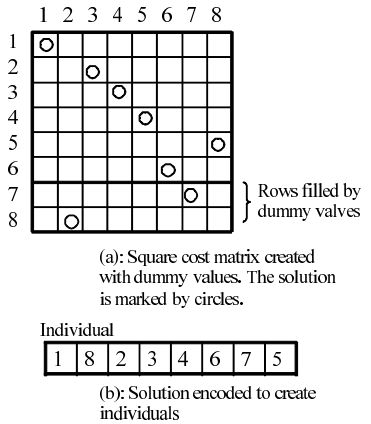

Fig. 8 Example of encoding candidate solution. The part of the cost matrix with dummy values is not employed in the calculation of fittness.

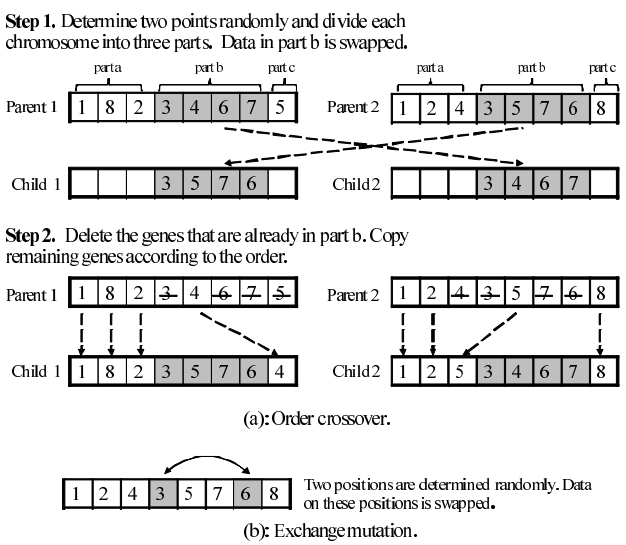

Fig. 9 Order crossover and exchange mutation.

4.2 Improvement of Genetic Algorithm

A normal GA generates initial individuals randomly to form an initial population [5]. In this study, we first solve the matching problem of Eq. (1). This result is employed for generating initial individuals.

It is still very difficult to search the approximate solution because the convergence becomes very slow when the size of cost matrix is huge. However, the smaller the size of block, the more details of the object in the current frame the matching can provide. In order to improve this problem, two kind of techniques are adopted. First, instead of searching all the blocks in the original cost matrix, we select M (the number of template blocks) blocks with top M similarities from every row of the basic dense cost matrix in Fig. 3 to generate a sparse cost matrix as shown in Fig. 5. Elements selected from the parts of current frames 2 and 3 are the same as those selected from current frame 1. Then, the GA is performed in the sparse cost matrix. Second, we employ an image pyramid method to perform template matching.

The flow of our proposed method is described as follows:

1. Scan the current frame and template image by block with block size 16×16 pixels and create the cost matrix.
2. Solve the matching problem of Eq. (1) as a linear assignment problem to obtain the optimal matching of the visual feature constraint. The solution is encoded to generate the initial individuals for the GA.
3. Create a sparse cost matrix. Perform the GA to search the approximate solution considering both the visual feature and structure constraints. Here, we assume that the extracted block set in the current frame is B_1.
4. Perform the morphological operation of dilation on B_1 and obtain a block set B_2 which is the candidate block set for the next step. This processing ensures that B_2 includes blocks of B_1 and their neighboring blocks.
5. Scan the area of B_2 and the template into one-dimensional data by block with block size 8×8 pixels. Create a cost matrix using these blocks.
6. Perform the GA to search the approximate solution considering both the visual feature and structure constraints.

5. Experimental Results

In this section, we report the experimental results of the proposed method.

5.1 Parameters of Experiments

In our experiments, we employ two groups of data. One is our data set containing outdoor scenes and the other is the benchmark data set from [19]. We normalize the Bhattacharyya distance in the range from 0 to 1000. All parameters are listed in Table 1. We determine the appropriate values of these parameters through experiments on data shown

Table 1 Parameters of experiments.

Image size	320 × 240	r of HOG Context histogram	3	Population size of GA	500
Block size	16 × 16 pixels/8 × 8 pixels	θ of HOG Context histogram	16	Crossover rate of GA	0.8
α in Eq. (2)	0.25	h of HOG Context histogram	9	Mutation rate of GA	0.5
β in Eq. (2)	0.25	Number of bins of HSV histogram	110	Number of generations of GA	200
w in Eq. (5)	30	Number of bins of HOG histogram	9		
K in Eq. (1)	3	Threshold for vanishing in Eq. (1)	600		
PC for experiments	Core 2 Duo 3.00 GHz PC with 2 GB RAM, WindowsXP				

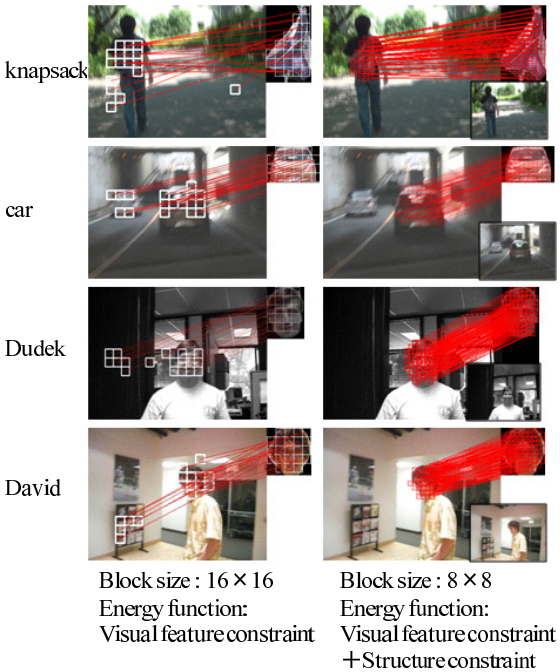

Fig. 10 Images on the left are results obtained by minimizing visual feature distances for block size 16 × 16 pixels. Images on the right are results obtained by minimizing both visual feature distances and structure distances by pyramid processing for block size 8 × 8 pixels.

in Figs. 11 (a) and (b). Then, the parameters are fixed for all experiments. The processing time is approximately 5 ∼ 10 s per frame. We aim to improve this processing time using parallel processing and a hardware accelerator in our future works. The proposed method is suiable for parallel processing for the following reasons:

1. Since the exclusive block matching uses fixed size blocks, parallel processing can be employed for almost all processes of blocks such as extracting features and calculating similarities.
2. An approximate method called saving-regret [20] is used for linear assignment problems. This method with high concurrency can realize high-speed processing.
3. Parallel processing is very efficient for GA because the processes for different individuals in a population can be performed simultaneously.

5.2 Experimental Results of Improvement by Optimizing both Visual Feature Distances and Structure Distances

First, we present some experimental results to indicate how the proposed method improves matching by considering both visual feature and structure constraints. Images on the left side of Fig. 10 show the results obtained by minimizing only visual feature distances for block size 16 × 16 pixels. Images on the right side show the final results obtained using our method for block size 8 × 8 pixels. These results indicate that inaccurate matching occurs frequently when only the visual feature distance is minimized. Figure 11 shows the process by which the GA improves matching results by minimizing both the visual feature and structure distances. The numbers on the left are generations in GA. Figure 12 shows the variations of fitnesses, i.e., the value E in Eq. (5). The extracted blocks of these results are then used to perform block matching again with block size 8 × 8 pixels.

5.3 Improvement of Convergence Using Sparse Cost Matrix

In this subsection, we employ the data of the knapsack image to make a comparison between the results obtained using the dense cost matrix and sparse cost matrix. As shown in Fig. 13, the fitness is reduced very slowly when the GA is performed for the dense cost matrix. It reaches 235748 in the end of the 1924th generation. On the other hand, the convergence of the GA is improved significantly when GA is performed for the sparse cost matrix. It reaches the same value of fitness in the end of the 651th generation. The fitness value of the 200th generation is 237596 which is only 100.8% of the value of the 651th generation. In order to maitain the balance between appropriate fitness and amount of calculation, we set the number of generations of GA to 200 for all experiments.

5.4 Comparison with SIFT Method and Mean-Shift Method

In this section, we present the results of the comparison of our method with the SIFT method and Mean-Shift method. The result of the Mean-Shift method is calculated using the OpenCV library [21]. The code of SIFT is obtained from Rob Hess's website [22]. We adjust the parameters to obtain the best possible results. According to the results of Fig. 14, the SIFT feature point could not be extracted from many areas of the object and there are many instances of inaccurate matching because the feature points are unstable when the illumination of the environment and the appearance of the target object change. The Mean-Shift method that calculates the distance on the HSV histograms is affected by

Fig. 11 Examples to illustrate the enhancement of the matching results using our method, considering both visual feature and structure constraints. The block size in this step is 16 × 16 pixels. The numbers on the left are generations in GA. Then extracted blocks of these results will be used to perform block matching again with block size 8 × 8 pixels.

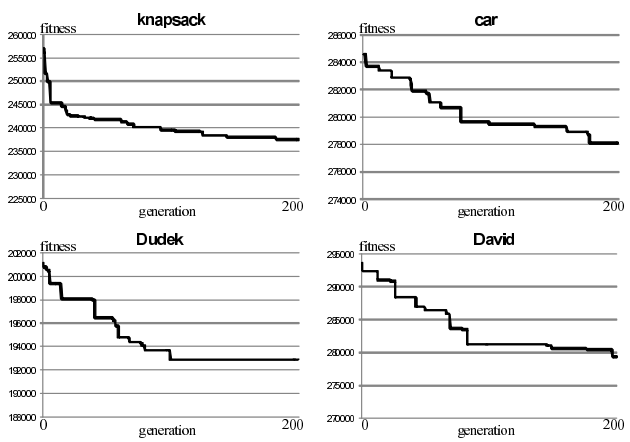

Fig. 12 Variation of fitness. The vertical axis represents the fitness and the horizontal axis represents the generation in GA. The block size is 16 × 16 pixels.

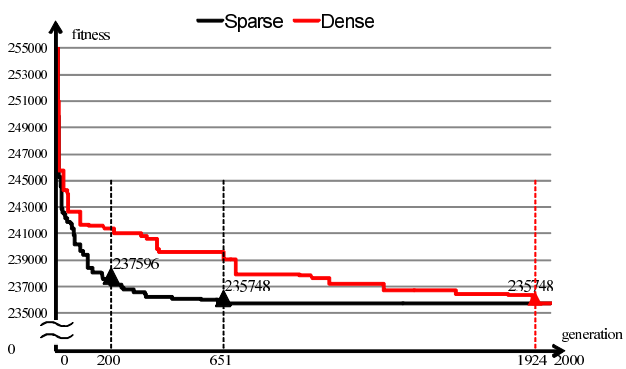

Fig. 13 Improvement of convergence by sparse cost matrix. The vertical axis represents the fitness and the horizontal axis represents the generation in GA. The block size is 16 × 16 pixels.

Fig. 14 Comparison of our method with the SIFT and Mean-Shift methods.

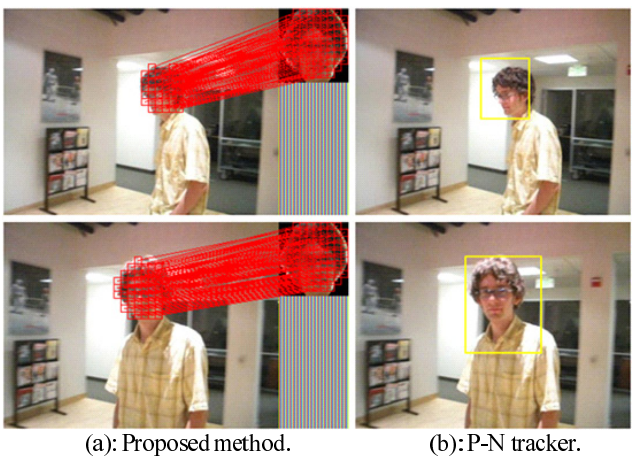

Fig. 17 Comparison with Kalal's method.

these problems as well. Our method realizes accurate tracking and obtains the corresponding relations between objects and templates.

5.5 Experimental Results of Outdoor Data and Benchmark Data

Finally, we present two groups of experimental results. One is the result of our data sets of real outdoor scenes including the tracking of a knapsack, two cars, and a person. There are many challenging issues in these image sequences. In Fig. 15 (a), illumination conditions change abruptly. In the data of Fig. 15 (b), there are many similar objects in scenes. Figures 15 (c) and (d) are the tracking results of image sequences captured by a hand-held camera. The images appeared blurred because of the movement of the camera. Moreover, the appearances of the target objects vary greatly. As can be observed from Fig. 15, our method realizes robust tracking and the corresponding relations between objects and templates are obtained accurately. Another group of data which is used for face tracking is downloaded from David Ross's website [19]. We test our proposed method by using his data and comparing our method with his method which is presented in [23]. The results are shown in Figs. 16 (a), (b), (c), and (d). Our method performs well even for the frames where Ross's method fails. Figures 16 (e) and (f) shows tracking results for another part of Ross's data. His results for this part are not presented; therefore, we only present the results obtained by our method. The proposed method tracks target objects accurately for these challenging image sequences as well.

Table 2 Comparison with recent tracking methods in terms of the frame number after which the tracker doesn't recover from failure. Results of these methods are presented in [27] and [28].

Frames	[24]	[25]	[26]	[27]	[28]	proposed method
761	17	94	135	759	**761**	**761**

We also compare the proposed method with recent tracking methods [24]–[28] using the David sequence in Table 2. Results of these methods are presented in [27] and [28]. These results show performances of P-N tracker [28] and the proposed method are best. Figures 17 (a) and (b) are some example frames of tracking results obtained by the proposed method and P-N tracker. Example frames of P-N tracker are obtained by using the demo program which is downloaded from Kalal's website [29]. According to Fig. 17, the proposed method shows higher precision and provides more details of the target object.

The image at the bottom of Fig. 16 (e) is an inaccurate tracking result. The resason is that the appearance of the target object varies greatly and the similarities of some blocks in the scene are higher than those between the template and the target object. In order to resolve this issue, we will use the location in the previous frame in our future work.

6. Conclusion

In this paper, we proposed a template matching method that considers both visual feature and structure constraints. This method is robust to many problems and can obtain the corresponding relations between objects and templates. In our future research, we will focus on improving the proccessing time through parallel processing and using a hardware accelerator. Moreover, we aim to introduce the spatio-temporal constraint to further enhance the robustness of the proposed method.

References

[1] D. Comaniciu, P. Meer, and S. Member, "Mean shift: A robust approach toward feature space analysis," IEEE Trans. Pattern Anal. Mach. Intell., vol.24, no.5, pp.603–619, 2002.
[2] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," European Conference on Computer Vision, pp.661–675, 2002.
[3] K. Nummiaro, E. Koller-Meier, and L.V. Gool, "A color-based particle filter," Image Vis. Comput., vol.21, no.1, pp.53–60, 2002.

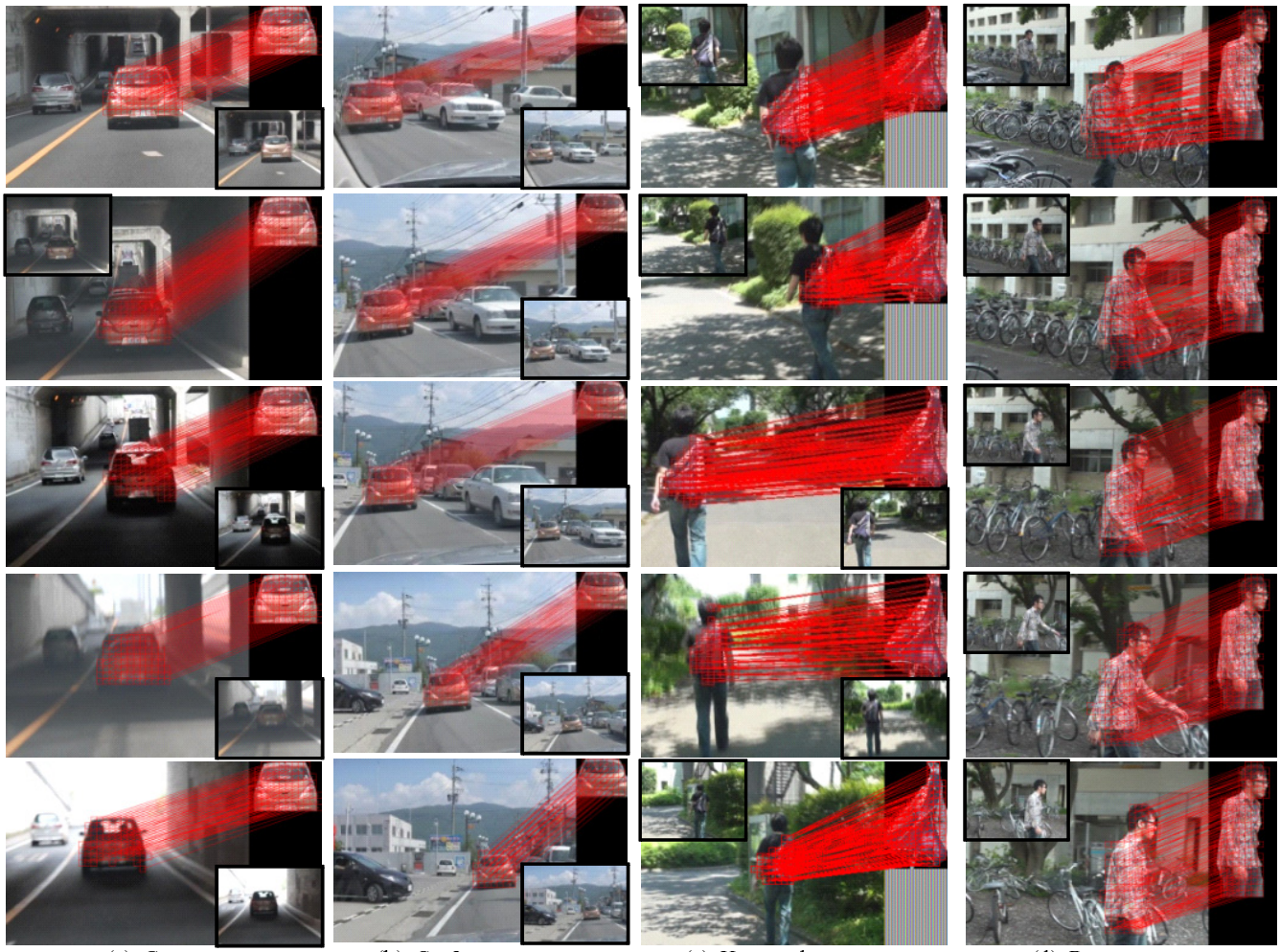

Fig. 15 Tracking results of outdoor real data.

[4] D. Lowe, "Object recognition from local scale-invariant features," Proc. International Conference on Computer Vision 2, pp.1150–1157, 1999.
[5] D.E. Goldberg, Genetic algorithms in search, optimization, and machine learning, Addison-Wesley, 1989.
[6] B.K.P. Horn and B.G. Schunck, "Determining optical flow," Artif. Intell., vol.17, no.1-3, pp.185–203, 1981.
[7] T. Brox, C. Bregler, and J. Malik, "Large displacement optical flow," IEEE International Conference on Computer Vision and Pattern Recognition, pp.41–48, 2009.
[8] T. Caetano, J. McAuley, L. Cheng, Q. Le, and A. Smola, "Learning graph matching," IEEE Trans. Pattern Anal. Mach. Intell., vol.31, no.6, pp.1048–1058, 2009.
[9] H. Jiang, S.X. Yu, and D.R. Martin, "Linear scale and rotation invariant matching," IEEE Trans. Pattern Anal. Mach. Intell., vol.33, no.7, pp.1339–1355, 2011.
[10] A.Y. Xin, X. Li, and M. Shah, "Object contour tracking using level sets," Asian Conference on Computer Vision, pp.1–7, 2004.
[11] C. Bibby and I. Reid, "Robust real-time visual tracking using pixel-wise posteriors," Proc. European Conference on Computer Vision, pp.831–844, 2008.
[12] K. She, G. Bebis, H. Gu, and R. Miller, "Vehicle tracking using on-line fusion of color and shape features," IEEE Int. Conf. on Intelligent Transportation Systems, pp.731–736, 2004.
[13] 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.
[14] "Munkres assignment algorithm." http://csclab.murraystate.edu/bob.pilgrim/445/munkres.html
[15] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," IEEE International Conference on Computer Vision and Pattern Recognition, pp.886–893, 2005.
[16] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," IEEE Trans. Pattern Anal. Mach. Intell., vol.27, no.10, pp.1615–1630, 2005.
[17] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," IEEE Trans. Pattern Anal. Mach. Intell., vol.24, no.4, pp.509–522, 2002.
[18] L. Davis, "Applying adaptive algorithms to epistatic domains," Proc. International Joint Conference on Artificial Intelligence, pp.162–164, 1985.
[19] "Data sets from David Ross." http://www.cs.toronto.edu/~dross/ivt/
[20] M.A. Trick, "A linear relaxation heuristic for the generalized assignment problem," Naval Research Logistics, vol.39, no.2, pp.137–151, 1992.
[21] "Opencv." http://opencv.jp/document/opencvref_cv.html
[22] "Code of sift keypoint detector." http://web.engr.oregonstate.edu/~hess/index.html
[23] D.A. Ross, J. Lim, R.S. Lin, and M.H. Yang, "Incremental learning for robust visual tracking," Int. J. Comput. Vis., vol.77, no.1-3,

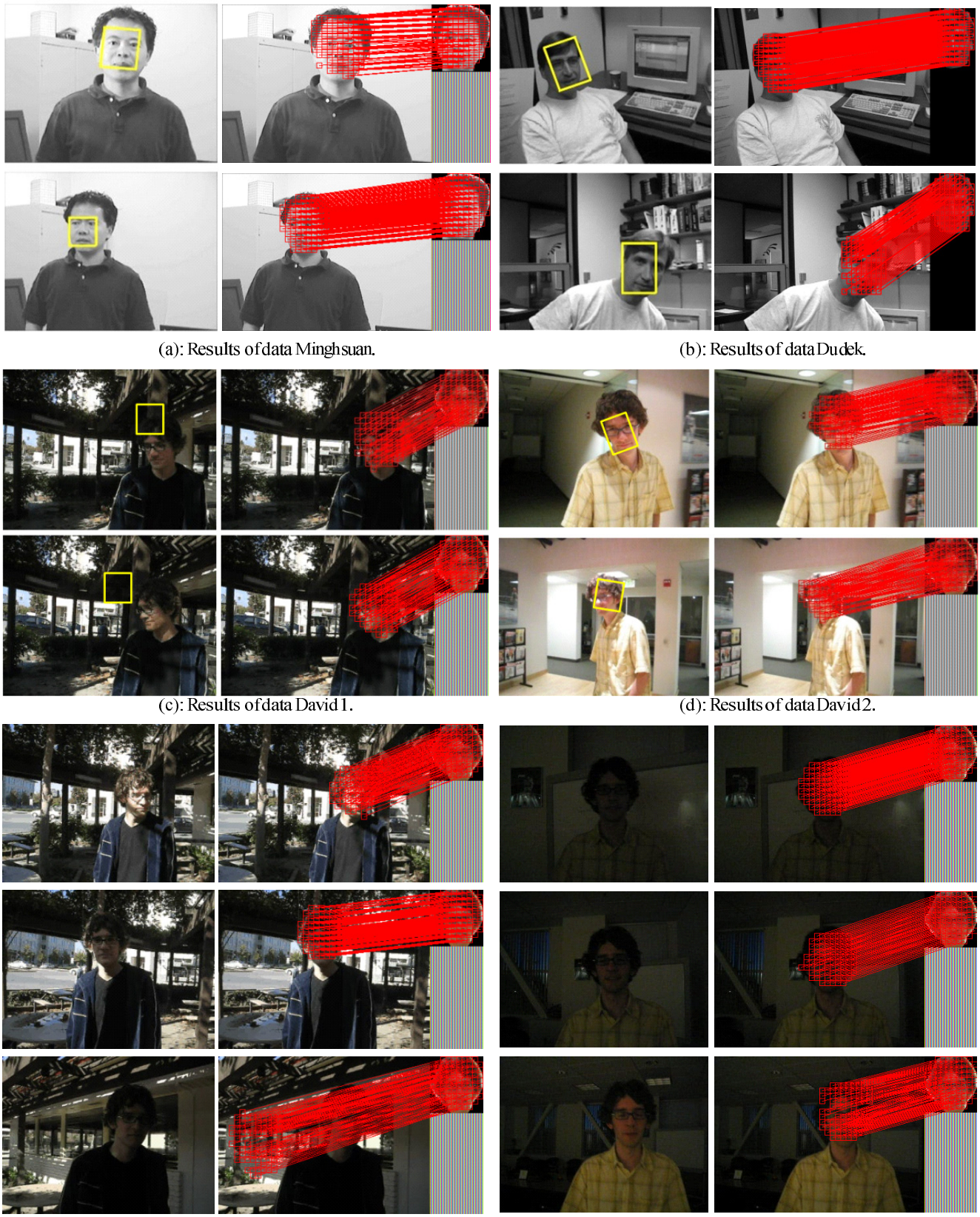

Fig. 16 Comparison with Ross's method. Images on the right are results of our proposed method. Images on the left of (a), (b), (c), and (d) are results of Ross's method. In (e) and (f), we only show our experimental results because his results for these frames are not presented. The proposed method tracks target objects accurately for most frames. In (e), inaccurate matching occurs because the appearance of the target object varies greatly and the similarities of some blocks in the scene are higher than those between the template and the target object.

pp.125–141, 2008.
[24] J. Lim, D. Ross, R.S. Lin, and M.H. Yang, "Incremental learning for visual tracking," Advances in Neural Information Processing Systems, pp.793–800, 2004.
[25] S. Avidan, "Ensemble tracking," IEEE Trans. Pattern Anal. Mach. Intell., vol.29, no.2, pp.261–271, 2007.
[26] B. Babenko, M.H. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," IEEE International Conference on Computer Vision and Pattern Recognition, pp.983–990, 2009.
[27] Y. Qian, D.T. Ba, and G.G. Medioni, "Online tracking and reacquisition using co-trained generative and discriminative," European Conference on Computer Vision, pp.678–691, 2008.
[28] Z. Kalal, J. Matas, and K. Mikolajczyk, "P-N Learning: Bootstrapping binary classifiers by structural constraints," Conference on Computer Vision and Pattern Recognition, pp.45–56, 2010.
[29] "Zdenek Kalal's homepage." http://info.ee.surrey.ac.uk/Personal/Z.Kalal/tld.html

Zhu Li received his B.S. degree in Electrical and Information Engineering from ZheJiang University of Technology, China, in 2005. Then, he received the M.S. degree in Electrical and Electronic Engineering from Tokyo University of Agriculture and Technology, Japan, in 2009. Currently, he is a Ph.D. student at the Department of Electronic and Information Engineering of Tokyo University of Agriculture and Technology. His research interests are image processing for object detection and tracking.

Kojiro Tomotsune received his B.S. degree in Electrical and Electronic Engineering from Tokyo University of Agriculture and Technology, Japan, in 2010. He is currently a master course student at the Department of Electrical and Electronic Engineering at the Tokyo University of Agriculture and Technology. 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.