

Real-Time Uncharacteristic-Part Tracking with a Point Set

Norimichi UKITA^{†a)}, Member, Akira MAKINO[†], Nonmember, and Masatsugu KIDODE[†], Fellow

SUMMARY In this research, we focus on how to track a target region that lies next to similar regions (e.g. a forearm and an upper arm) in zoom-in images. Many previous tracking methods express the target region (i.e. a part in a human body) with a single model such as an ellipse, a rectangle, and a deformable closed region. With the single model, however, it is difficult to track the target region in zoom-in images without confusing it and its neighboring similar regions (e.g. “a forearm and an upper arm” and “a small region in a torso and its neighboring regions”) because they might have the same texture patterns and do not have the detectable border between them. In our method, a group of feature points in a target region is extracted and tracked as the model of the target. Small differences between the neighboring regions can be verified by focusing only on the feature points. In addition, (1) the stability of tracking is improved using particle filtering and (2) tracking robust to occlusions is realized by removing unreliable points using random sampling. Experimental results demonstrate the effectiveness of our method even when occlusions occur.

key words: real-time tracking, zoom-in camera, point-set tracking

1. Introduction

High-resolution images of a target object can improve the performance of various existing image-analysis algorithms. In particular, high-resolution images of human regions (e.g. a face, a hand, a forearm, and a tiny part of them as shown in Fig. 1) are useful in order to observe facial expression, hand/finger motion, and other minute motions and features. In a security system (e.g. [1]), for example, actions of a suspicious individual can be observed circumstantially and recognized well even by employing existing algorithms.

Continuous high-resolution imaging of a moving target can be realized by using a pan-tilt-zoom (PTZ) camera. For zoom-in observation with the PTZ camera, the following two techniques are required:

Tracking Continuously extract the region of the target in observed images.

Camera-control Continuously control the PTZ camera for capturing the target within the images.

If tracking is successful, a previously proposed camera-control algorithm (e.g. [2], [3]) is applicable to continuous high-resolution imaging regardless of the type of the target. We, therefore, focus on how to track a partially observed region of a moving object in real time instead of observing the whole image of the object (Fig. 1).

When observing the whole body of a person, each body part can be easily identified by employing a whole-body model (see [4], [5], for example). This approach is, however, inapplicable to zoom-in observation because the whole body is not observed. Furthermore, appearance information such as texture patterns is not known in advance because of a variation of clothing.

In this research, we have to employ a tracking algorithm for general objects, which can work without prior knowledge with regard to a target object. Template matching is widely used for tracking (see [6], for example). Although it is applicable to any object tracking by extracting the template in an initial frame, the similarity can be high not only in the correct region of a target but also in its neighboring regions with textures similar to those of the correct region as shown in Fig. 2. In addition, it should be noted that the target region might be non-rigid (e.g. body parts with loose-fitting clothing and multiple limbs). Tracking using SNAKE [8], which searches for the boundary line of a target based on edge lines and their smoothness, is also inapplicable to tracking body regions with similar textures between which there is no edge line. Mean-shift [9] is also one of the famous real-time tracking algorithms. While this method is

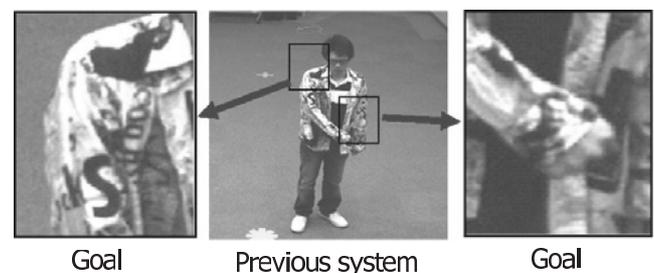


Fig. 1 Goal of this work: zoom-in tracking of a moving part.

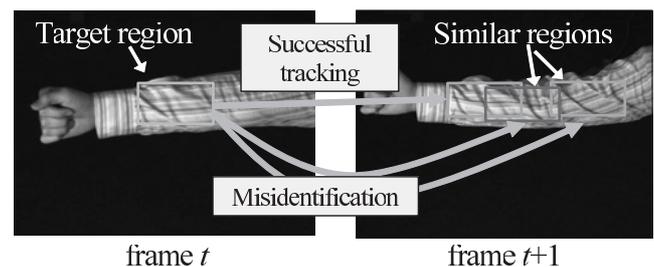


Fig. 2 Misidentification of a target region.

Manuscript received October 8, 2009.

Manuscript revised January 2, 2010.

[†]The authors are with Nara Institute of Science and Technology, Ikoma-shi, 630-0192 Japan.

a) E-mail: ukita@is.naist.jp

DOI: 10.1587/transinf.E93.D.1682

robust against deformation of a target shape, mistracking between similar neighboring regions is not avoidable because this method employs only a color histogram. As described above, it is difficult to express the difference between similar neighboring regions by employing a single model representing the whole region of a target.

For zoom-in tracking, in this research, a set of feature points in a target region is regarded as a tracking target. The difference between similar neighboring regions can be expressed well by focusing only on characteristic feature points. In addition, by tracking the set of the points, (1) an arbitrary object can be tracked, (2) an occluded target region can be also tracked by finding other visible points, and (3) small deformation of the target region can be represented by the change in the geometric configuration of the points.

Point tracking is one of major problems (e.g. for stereo vision and shape from motion) in Computer Vision. In [10], for example, the Lucas-Kanade algorithm [11] is analysed in detail in terms of computational complexity and stability. Tracking robust to specular highlights and lighting changes [12] is also important. Our method proposed in this paper is peculiar in terms of the following two characteristics:

Accept small drifts: The objective of most of point tracking algorithms is to determine the precise position of each point (e.g. for 3D reconstruction). A small *drift* of each point is, on the other hand, acceptable in our objective. This is because points might be included in the target region even when small drifts occur. In this research, therefore, the stability of tracking should be improved in exchange for the decline of precision. We employ particle filtering [13] for robustly tracking feature points with updating template images. Particle filtering and template update allow us to increase the stability and continuity of tracking, respectively.

Explicitly remove outlier points: As with our approach, particle filtering is used to track a set of points in [16]. In [16], however, all points are evaluated equivalently for calculating the likelihood of each particle. This results in tracking failure when one or more of feature points are invisible due to occlusions or located far from their true positions due to invalid sampling. That is, these outlier points cause mistracking, unlike small drifts mentioned above. This problem can be solved by removing these occluded points from each particle.

2. Tracking with Particle Filtering

This section introduces the outline of tracking with particle filtering [13]. In particle filtering, a target region is represented as a state vector. A set of particles, each of which corresponds to a state vector, is distributed in order to find the target region.

Let $\{P^1(t), \dots, P^N(t)\}$ and $\{\rho^1(t), \dots, \rho^N(t)\}$ be the set of N particles at time t and their likelihoods, respectively. With them, the target region at t is determined and the particles at $t + 1$ are generated as follows:

1. **Likelihood calculation:** Likelihood $\rho^i(t)$ of $P^i(t)$ in an image observed at time t is calculated.
2. **State estimation:** The region of the target in the observed image is estimated from the state vectors of the particles and their likelihoods (e.g. the weighted mean the state vectors).
3. **Sampling:** New particles $P^i(t + 1)$ are generated based on the likelihoods of the particles at t .
4. **Drift:** $P^i(t + 1)$ are shifted based on the motion dynamics of the target.

With the above mentioned scheme, robust tracking can be achieved. It should be noted that the computational cost increases drastically if the number of the particles is increased for stable tracking. Therefore, the objective in this paper is to improve the stability of tracking with a limited number of the particles.

3. Feature Points Tracking with Particle Filtering

The basic scheme of our method is based on tracking with particle filtering [13] described in Sect. 2. The characteristic issue of our method is how to calculate the likelihood of each particle. Its process flow is shown in Fig. 3. The distinctive feature of our method is outlier elimination.

Assume that the region of a target is given at an initial frame. At this frame, several feature points in the initial region are extracted by using the Harris operator [17][†]. A set of these points is regarded as a target being tracked. The state vector P^i of i -th particle is expressed with image coordinates of all the points, denoted by $(x_1^i, y_1^i), \dots, (x_p^i, y_p^i), \dots, (x_{N_i}^i, y_{N_i}^i)$, where (x_p^i, y_p^i) and N_i

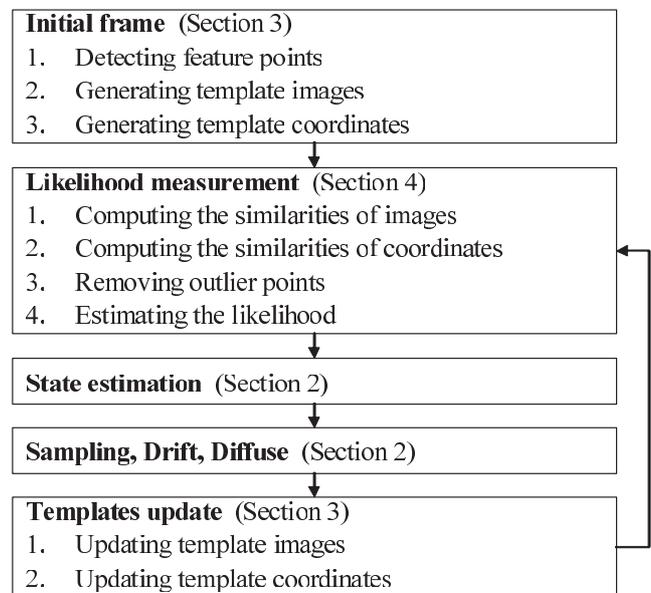


Fig. 3 Process flow of feature points tracking.

[†]We also conducted experiments using feature points extracted by [18]. The results of tracking were almost same between both methods.

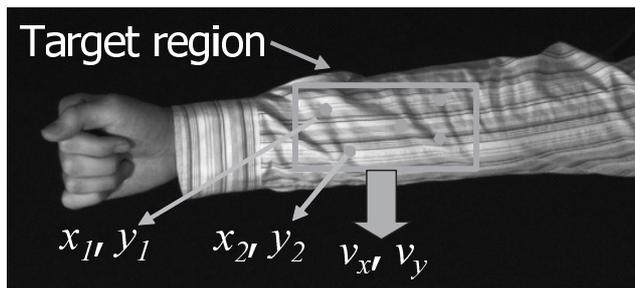


Fig. 4 Definition of a particle.

denotes xy coordinates of p -th point in i -th particle and the number of points in i -th particle, respectively, and their average velocity, (v_x^i, v_y^i) , as follows (Fig. 4):

$$P = [x_1^i, y_1^i, \dots, x_p^i, y_p^i, \dots, x_{N_i}^i, y_{N_i}^i, v_x^i, v_y^i] \quad (1)$$

On the other hand, particle filtering can be achieved also so that *each feature point* is regarded as a tracking target while the approach proposed in this paper deals with a set of the points as a target. For tracking multiple targets, the mixture particle filter [14] is superior to the original one [13]. The mixture particle filter maintains the models (i.e. probability distribution) of the targets in order to track them separately. In [15], the mixture particle filter is integrated with Adaboost for efficient object detection and occlusion recovery. This property, however, cannot be employed in our problem because neither to train target samples in advance nor to train them online in real time is possible. Moreover, the independent maintenance of the multiple models is not effective for our problem because no geometric assumption among the feature points is employed; all feature points on a target region move with the same motion dynamics in our problem. Our method explicitly employs this assumption for robust tracking.

Assume that the points have the following natures:

- The change in the local image around a feature point is small at a small interval.
- The change in the geometric configuration of feature points is small at a small interval.

With these natures, two templates below are used:

Template image Local image around each point.

Template coordinates xy coordinates of each point.

These templates are obtained at an initial frame and they are dynamically updated. The likelihood of each particle is calculated by integrating the following two values (described in detail in Sect. 4):

Image similarity Similarity between the local image around each point in a particle and the template image of the point.

Coordinates similarity Similarity between the geometric configurations of the points in the template coordinates and each particle.

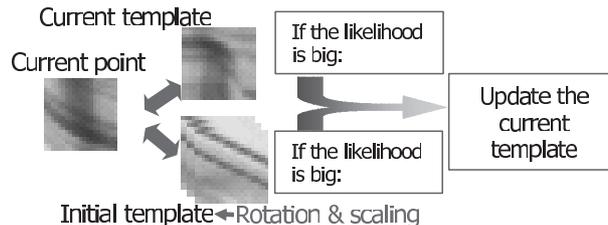


Fig. 5 How to update a template image.

Note that even the sampled point nearest to the true position might be a little different from the true position. This difference results in the significant decline of the image similarity. This negative effect is relaxed with a weighted evaluation that gives a high similarity to a point near the true position as described in Sect. 4.1. This weighted evaluation allows us also to cope with the small deformation of points. Although non-rigid tracking can be successful based on constraints with regard to 3D object/motion (e.g. [19]), this approach requires large computational cost. For example, in order to control a PTZ camera depending on target motion, real-time tracking is desired (e.g. 30 fps). In this work, we aim at real-time tracking for small but random deformation (e.g. deformation of clothing).

Unlike the effects due to small drift and deformation, outlier points must be removed for successful tracking. In our method, the outlier points are eliminated by random sampling as described in Sect. 4.4. This process enables real-time tracking robust against occlusions and other errors in point tracking.

As with the previous tracking algorithm using particle filtering [13] described in Sect. 2, the weighted mean of the particles is regarded as the state of the target (i.e. the estimated points). Sampling and drift of our algorithm are also similar to those of [13]. The estimated velocity of the target (i.e. v_x and v_y in a particle) is used to shift the particles in the current frame.

Although the appearance of the target object in observed images change as it moves, careless update of the templates incurs mistracking. In our method, therefore, the image templates are updated not only using the updated template but also using the initial template as proposed in [6]. The template image is updated if both of the following two conditions are satisfied [6] as illustrated in Fig. 5: (1) the similarity between the current template image and the local image around the estimated point is higher than a threshold and (2) the maximum similarity between the local image around the estimated point and the images that are obtained by rotating, translating, expanding, and shrinking the initial template image is higher than a threshold. It should be noted that the weight of matching with the initial template image is smaller than that with the current template image because small drift is acceptable in our objective. The template coordinates are also updated in the same way with updating the template image.

4. Image Likelihood

This section describes 1) how to compute the likelihood of each particle with two similarities in Sect. 4.1, 4.2, and 4.3 and 2) likelihood computation with outlier removal that overcomes small deformation of points and occlusion in Sect. 4.4.

4.1 Similarity of Feature Points

The image similarity is evaluated by the SAD (Sum of Absolute Difference). The computed SAD is normalised (i.e. from 0 to 1) by the template size and the maximum value of the intensity range.

Note that even the position of each feature point in the best particle might be a little different from the true position. Since even the small difference results in the significant decrease of the image similarity (i.e. SAD), the similarity of the best point might be accidentally lower than that of another point. To cope with this problem, the similarity is determined so that the similarity decreases drastically at some distance from the true position as illustrated in Fig. 6. The image similarity of point p in particle i , I_p^i , is calculated by the following equation:

$$I_p^i = 2v \sin\left(\frac{\pi \cdot 0.5}{2h} c_p^i\right) \quad \text{if } c_p^i \leq h \tag{2}$$

$$I_p^i = 2(1 - v) \sin\left(\frac{\pi}{2}(c_p^i - (2h - 1))\right) + (2v - 1) \quad \text{if } c_p^i > h \tag{3}$$

where $c_p^i = 1 -$ “the normalised SAD”, and h and v are constants[†] indicated in Fig. 6.

4.2 Similarity of Geometric Configurations

The similarity of geometric configurations is calculated by comparing the template coordinates and the coordinates of

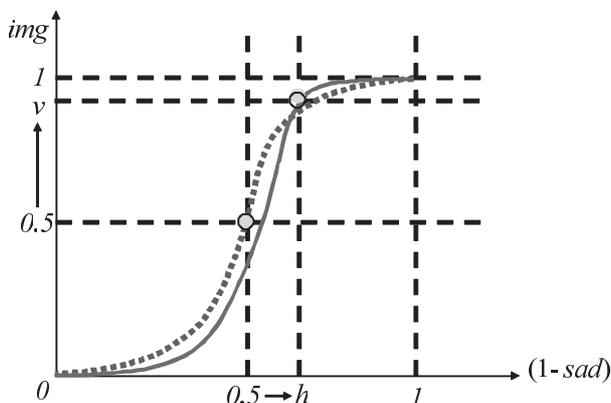


Fig. 6 Weighted similarity function (dotted line: sin curve, solid line: a weighted similarity).

points in each particle. As illustrated in Fig. 7, let P_p^i and T_p be “the coordinates of point p in particle i ” (i.e. (x_p^i, y_p^i) in Eq. (1)) and “the coordinates of point p in the template coordinates”, respectively. Note that the correspondences of points between P_p^i and T_p is known because the ID of each point (i.e. p) (i.e. the geometric configurations of the points) is known in particle i ; no image feature matching is needed.

Even if the point correspondences are known, P_p^i is translated, rotated, and deformed from T_p . The coordinates similarity is determined based on how large the deformation is. To evaluate the deformation, P_p^i and T_p are aligned in terms of the translation and rotation.

For the alignment, the affine parameters A^i between P_p^i and T_p is estimated, and then T_p is transformed using A^i as illustrated in the right-hand of Fig. 7. In [7] also, a target region is tracked while aligning it with its appearance model, which is leaned online. As achieved in this tracking method, most of similar methods deal with a small region though a set of points are employed for tracking robust to occlusion in our method.

The distance dis_p^i between the transformed coordinates $A^i T_p$ and P_p^i is obtained by $dis_p^i = |A^i T_p - P_p^i|$. With dis_p^i , the coordinates similarity C_p^i is expressed by the following formula:

$$C_p^i = \frac{1}{dis_p^i + 1} \tag{4}$$

4.3 Likelihood with Image and Coordinates Similarities

Using the image and coordinates similarities I_p^i and C_p^i , the likelihood of particle i (denoted by ρ^i) is estimated by the formula below:

$$\rho^i = \left(\sum_{p=1}^{N^i} e_p^i \right) / N^i, \tag{5}$$

where N^i is the number of the feature points in i -th particle and $e_p^i = I_p^i \cdot C_p^i$, which denotes the similarity of point p in particle i .

4.4 Removing Outlier Points

The affine parameters A^i is easily estimated by the method described in Sect. 4.1, 4.2, and 4.3 if all points in particle i are observable and not deformed significantly from T_p . This assumption is often violated due to undesired motion of a

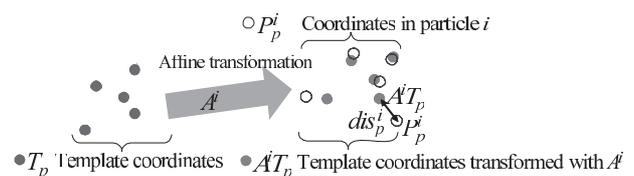


Fig. 7 Affine transformation of points.

[†]In our experiments, $h = 0.8$ and $v = 0.9$.

target region. This violation should be detected for robust computation of A^i .

Our method detects the violation of the assumption based on steep falloff of similarity e_p^i . e_p^i becomes very low in positions occluded by another object or far from its true position as illustrated in Fig. 8 (a) and (b), respectively. We call such a point an *outlier point*. The outlier point results in a low likelihood of the corresponding particle even if all other points have high evaluated values. In general, the former problem (i.e. a wrong position) can be easily resolved by distributing a large number of particles in order to prepare the one in which all points are located in their correct positions. As the particles increase, however, the computational speed declines. Furthermore, the latter problem (i.e. an occluded point) cannot be solved by boosting the particles. In our method, therefore, the likelihood of a particle is evaluated without points, each of which has an extremely low value.

In our method, the positions of feature points in each particle are evaluated by comparing them with the points transformed from the template coordinates by an affine transformation. The affine parameters can be estimated from at least three corresponding points. If one or more outlier points are included in the corresponding points, the estimated affine parameters produce the transformed points, all of which are far from the points in the particle. If no outlier point is included, e_p^i may be low only in the outlier point(s). Outlier removal is, therefore, implemented by robust estimation (least median square estimation [20]) with random sampling as follows:

Step 1 Select random three points in a particle and estimate the affine parameters \hat{A}^i between them and the template coordinates.

Step 2 Transform the template coordinates using \hat{A}^i .

Step 3 Calculate e_1^i, \dots, e_N^i and select their median value, e_{med}^i .

Step 4 Repeat Steps 1, 2, and 3 k times and consider the maximum e_{med}^i to be the one that corresponds to the optimal affine parameters A^i .

Steps 1, 2, and 3 should be repeated so that no outlier point is included in at least one combination of the selected points. The probability that at least one combination without outliers is selected (denoted by r)[†] is determined [20] by $r =$

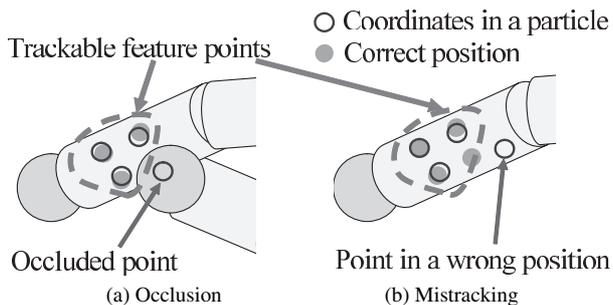


Fig. 8 Examples of outlier points.

$1 - \{1 - (1 - \epsilon)^F\}^k$, where ϵ and F denote the ratio of outlier points in all the feature points and the number of selected samples (i.e. three in our case), respectively.

Finally, outlier points are removed using the optimal affine parameters A^i as follows:

Step 1 Calculate e_1^i, \dots, e_N^i using A^i .

Step 2 Compute the standard deviation of e_1^i, \dots, e_N^i (denoted by σ).

Step 3 Remove points, each of whose e_p^i is at a distance of over 2.5σ from the average, from particle i and then evaluate its likelihood.

5. Experiments

We conducted experiments using a PC (Pentium4 3.0 GHz) with 1024×768 pixels images captured at 30 fps (Pointgrey XGA Flea). In all the experiments, the number of particles, the number of points in a particle, and the size of a template image were 800, 7, and 15×15 , respectively. With these conditions, our method ran at 30 fps.

For comparative experiments, the following four methods were examined:

Method1 Template matching with SAD evaluation.

Method2 KLT tracker [10].

Method3 Points tracking using particle filtering WITHOUT removing outlier points.

Method4 Proposed method: Points tracking using particle filtering WITH removing outlier points.

One of the advantages of our method is to be able to track a target region neighboring similar regions as shown in Fig. 9. To confirm this advantage, a part of an arm with white clothing was tracked by four methods. The partial sequences of the results are shown in Fig. 9. The capturing interval of the images in Fig. 9 is 1/30 sec. The target region is enclosed by a rectangle. The results of Method1 stayed at the same area in the observed images while the arm translated. This is because the similarity is almost unchanged around the area. Method2 and Method3 lost track of the target region in midstream. Our proposed method could track the target region until the last frame.

The second advantage of our method with outlier removal is to be able to track a region partially occluded. As shown in Fig. 10, our method tracked a target region (an arm in an upper sequence and a face in a lower sequence) even when it was partially occluded. In the arm sequence, three points were occluded by the other hand at (a3), but successfully tracked at the following frames (a4) and (a5). In the face sequence, two points located in the left-upper corner of the face were not observed at (f3), but successfully tracked at the following frames (f4) and (f5). It can be seen that the observable points were tracked correctly even during the frames with these difficulties (e.g. (a3) and (f3)); compare these points at (a3)/(f3) and other frames.

[†]In our experiments, r was 0.95.

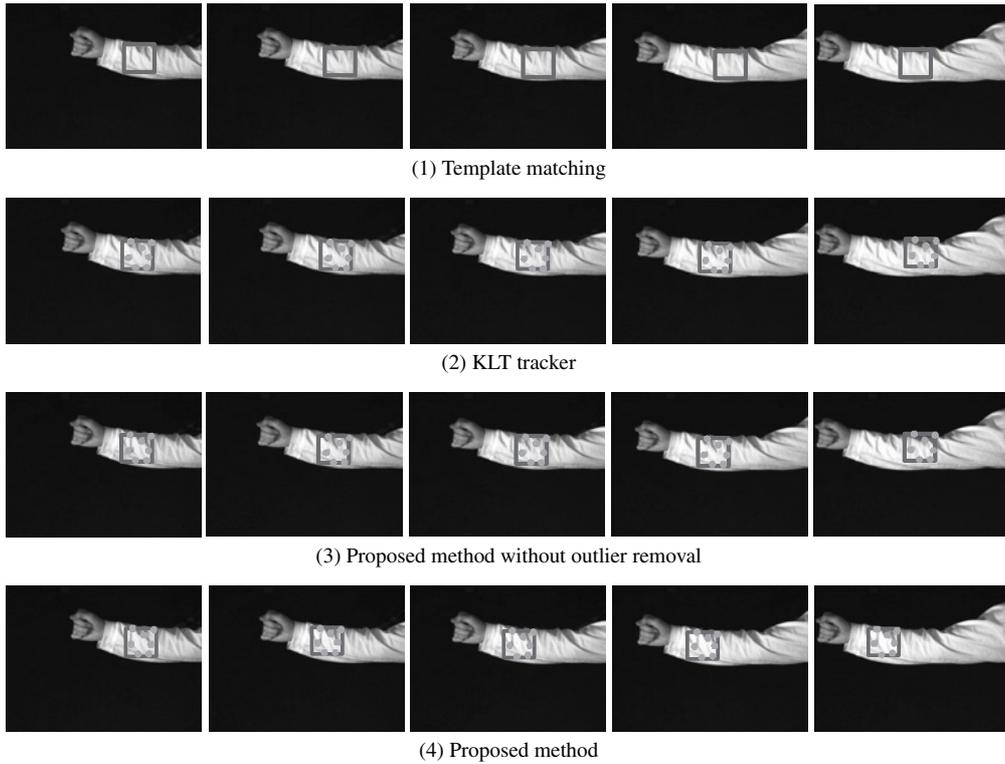


Fig. 9 Results of uncharacteristic part tracking.

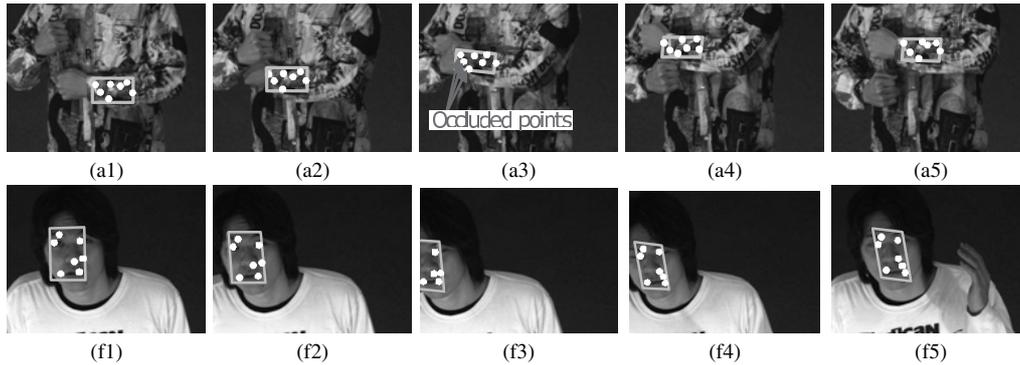


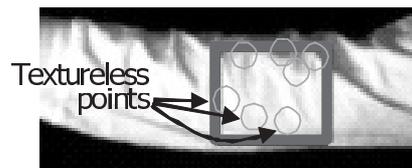
Fig. 10 Results of partially-occluded part tracking.



Fig. 11 Unsuccessful tracking results: Large deformation.



(a) Tracking results



(b) Magnified images of textureless feature points

Fig. 12 Unsuccessful tracking results: Points with no textures.

Figures 11 and 12 show examples of unsuccessful tracking. In Fig. 11, since an elbow was bent, i) the geometric configuration of feature points was changed and ii) textures were changed around several feature points that were made by folds. Figure 12 shows the example of textureless points. Figure 12 (b) shows the textureless points at the first frame. These points made it difficult to successfully track a

Table 1 Quantitative comparison: success rate of tracking.

	1. white shirt	2. textured shirt	3. face	4. large deformation	5. few folds
Method1	64.6%	72.0%	84.2%	46.2%	55.8%
Method2	73.6%	92.6%	88.8%	79.6%	67.6%
Method3	81.3%	88.8%	88.0%	55.2%	58.0%
Method4	86.8%	92.4%	93.2%	71.2%	65.4%

target region with a set of all points as shown in Fig. 12 (a).

We conducted experiments for quantitative comparison (Table 1). For five targets (1. a white shirt (Fig. 9), 2. a textured shirt (upper sequence in Fig. 10), 3. a face with partial occlusions (lower sequence in Fig. 10), 4. a textured shirt with large deformation (Fig. 11), and 5. a region with few folds on the white shirt (Fig. 12)), five sequences consisting of 50 frames were prepared. True regions of the target were given by hand and compared with the tracking results; a result overlapping over 80% with the true region is regarded as a successful one.

Our method was superior to the others in tracking the white shirt and the face with occlusions. With the textured shirt, successful tracking was achieved also by the KLT tracker because observable points could be tracked even during the difficult frames. With the shirt with large deformation, the point set should be divided into several groups as will be described in Sect. 6. With the region with few folds on the white shirt, it was difficult to extract the required number (i.e. 7 in these experiments) of sufficiently textured points. To resolve this problem, the number of the feature points should be determined in accordance with each target region.

6. Concluding Remarks

We focused on how to track an uncharacteristic region and proposed a points tracking method using particle filtering with removing outlier points.

Future work includes the following aspects:

Template update Our method employs the template update algorithm proposed in [6]. This algorithm compares not only a current template but also an initial template with a current image in order to avoid *drift*. Employing the initial template, however, prevents matching with a significantly deformed image. A more sophisticated way is, therefore, required for robust and high-speed tracking. For example, [22] proposed this kind of model update based on confidence of each tracking part.

Grouping If points are distributed in multiple limbs, the motions of the points differ substantially from that represented by an affine transformation. Then tracking might be unsuccessful. To cope with this problem, the points must be grouped based on their motions, as with the rigid motion based grouping algorithm [21].

Speed-up While real-time tracking can be achieved in our current implementation, the above mentioned processes would increase the computational cost. In the

current implementation, removing outlier points requires substantial time. To reduce its computational cost, the likelihood might be computed by a simpler way in the outlier removal.

Robustness to change in illumination A general tracking algorithm is required to be robust to change in illumination. While this property has not been validated in this paper, our method has ability to cope with partial change in illumination in principle; even if illumination conditions on several feature points are changed (e.g. due to cast shadows), these points can be regarded as outliers. In addition to validation of this advantage, robustness to more general changes in illumination should be achieved.

References

- [1] M.M. Trivedi, T.L. Gandhi, and K.S. Huang, "Distributed interactive video arrays for event capture and enhanced situational awareness," IEEE Intelligent Systems, vol.20, no.5, pp.58–66, 2005.
- [2] P.Y. Oh and P.K. Allen, "Design of a partitioned visual feedback controller," Proc. IEEE International Conference on Robotics and Automation, pp.1360–1365, 1998.
- [3] H. Wu, Q. Chen, H. Oike, C. Hua, T. Wada, and T. Kato, "High performance object tracking system using active cameras," Demo Proc. International Conference on Computer Vision, 2005.
- [4] T.B. Moeslund and E. Granum, "A survey of computer vision-based human motion capture," Computer Vision and Image Understanding, vol.81, no.3, pp.231–268, 2001.
- [5] R. Poppe, "Vision-based human motion analysis: An overview," Computer Vision and Image Understanding, vol.108, no.2, pp.4–18, 2007.
- [6] I. Matthews, T. Ishikawa, and S. Baker, "The template update problem," IEEE Trans. Pattern Anal. Mach. Intell., vol.26, no.6, pp.810–815, 2004.
- [7] S. Hinterstoisser, O. Kutter, N. Navab, P. Fua, and V. Lepetit, "Real-time learning of accurate patch rectification," Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2009.
- [8] N. Peterfreund, "The velocity snake: Deformable contour for tracking in spatio-velocity space," Computer Vision and Image Understanding, vol.73, no.3, pp.346–356, 1999.
- [9] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," IEEE Trans. Pattern Anal. Mach. Intell., vol.25, no.5, pp.564–575, 2003.
- [10] S. Baker and I. Matthews, "Lucas-Kanade 20 years on: A unifying framework," Int. J. Comput. Vis., vol.56, no.3, pp.221–255, 2004.
- [11] B.D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," Proc. Imaging understanding workshop, pp.121–130, 1981.
- [12] M. Gouffes, C. Collewet, C. Fernandez-Maloigne, and A. Tremeau, "Feature points tracking: Robustness to specular highlights and lighting changes," Proc. 9th European Conference on Computer Vision, vol.4, pp.82–93, 2006.
- [13] M. Isard and A. Blake, "Condensation—Conditional density propagation for visual tracking," Int. J. Comput. Vis., vol.28, no.1, pp.5–

- 28, 1998.
- [14] J. Vermaak, A. Doucet, and P. Pirez, "Maintaining multi-modality through mixture tracking," Proc. International Conference on Computer Vision, 2003.
 - [15] K. Okuma, A. Taleghani, N.D. Freitas, J.J. Little, and D.G. Lowe, "A boosted particle filter: Multitarget detection and tracking," Proc. European Conference on Computer Vision, pp.28–39, 2004.
 - [16] K. Kawamoto, "Guided importance sampling based particle filtering for visual tracking," Proc. IEEE Pacific-Rim Symposium on Image and Video Technology, pp.158–167, 2006.
 - [17] C.J. Harris and M. Stephens, "A combined corner and edge detector," Proc. 4th Alvey Vision Conference, pp.147–151, 1988.
 - [18] J. Shi and C. Tomasi, "Good features to track," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.593–600, 1994.
 - [19] L. Torresani, D.B. Yang, E.J. Alexander, and C. Bregler, "Tracking and modeling non-rigid objects with rank constraints," Proc. IEEE Conference on Computer Vision and Pattern Recognition, vol.1, pp.493–500, 2001.
 - [20] P. Rousseeuw and A. Leroy, Robust Regression & Outlier Detection, John Wiley & Sons, 1987.
 - [21] K. Inoue and K. Urahama, "Separation of multiple objects in motion images by clustering," Proc. International Conference on Computer Vision, vol.1, pp.219–224, 2001.
 - [22] B. Han and L.S. Davis, "Probabilistic fusion-based parameter estimation for visual tracking," Computer Vision and Image Understanding, vol.113, no.4, pp.435–445, 2009.



Norimichi Ukita received the B.E. and M.E. degrees in information engineering from Okayama University, Japan, in 1996 and 1998, respectively, and the Ph.D degree in Informatics from Kyoto University, Japan, in 2001. After working as an assistant professor at Nara Institute of Science and Technology (NAIST), he became an associate professor in 2007. He was a research scientist of PRESTO, Japan Science and Technology Agency (JST) from 2002 to 2006. He is now working at the Robotics Institute,

Carnegie Mellon University as a visiting research scientist. His main research interests are object detection/tracking and pose/shape estimation of human body and clothing. He has received the best paper award from the IEICE in 1999.



Akira Makino got his M.E. degree in engineering from Nara Institute of Science and Technology, Japan, in 2007. The theme of his master thesis was visual object tracking. He is now working at Nissan Motor Company.



Masatsugu Kidode received the B.E., M.E., and Ph.D degrees in engineering from Kyoto University, Japan, in 1968, 1970, and 1980, respectively. After working at Toshiba Corporation during 1970 and 1999, he joined at the graduate school of information science, Nara Institute of Science and Technology (NAIST), Japan in 2000 as a professor. He is a fellow of the following societies: IEEE, IAPR, and IPSJ. He served as general chair of IEEE ISWC200 and general chair of IAPR MVA2002. His research

interests include image processing, robot vision, and intelligent human interfaces.