Occlusion-Free Hand Motion Tracking by Multiple Cameras and Particle Filtering with Prediction

Makoto Kato[†] *, and Gang Xu*[†]

[†]Department of Media Technology, Ritsumeikan University, Kusatsu, Japan

Summary
This paper proposes a new technique to simultaneously estimate the global hand pose and the finger articulation imaged by multiple cameras. Tracking a free hand motion against a cluttered background is a difficult task. The first reason is that hand fingers are self-occluding and the second reason is the high dimensionality of the problem. In order to solve these difficulties, we propose using calibrated multiple cameras and at the same time improving search efficiency by predicted particle filtering. Therefore our methods can cope with both rapid global hand motion and self-occlusion. We also add prediction to particle filtering so that more particles are generated in areas of higher likelihood, which reduces search cost significantly. The effectiveness of our method is demonstrated by tracking free hand motions in real image sequences.

Key words:
Articulated hand tracking, Motion, Gesture recognition, Particle filtering, Motion capture.

Introduction

Recently, hand gesture recognition and hand motion tracking have become important issues in the field of human-computer interaction. Many vision-based approaches have been proposed [1]-[7] [14] [17].

The hand tracking methods by vision can be divided into two categories. One is appearance-based, and the other is model-based. In the appearance-based methods, mapping between image features and hand pose is established first, and hand pose estimation is formulated as an image database indexing problem, where the closest matches for an input hand image are retrieved from a large database of synthetic hand images [2]. The problem with the appearance-based method is the requirement of a very large database.

In contrast, the model-based methods use an articulated hand model. The hand pose at the current frame is estimated from the current image input and previous pose. The problem of using a hand model is the high dimensionality. The high dimensionality causes an exponentially high computational cost. Particle filtering is one of the most successful object tracking algorithms [9] [10]. However, to keep tracking correctness especially for rapid motions, it needs a large number of particles.

Another problem with the model-based approach is self-occlusion. While a hand moves freely, parts of the hand change from being visible to being invisible, and then becoming visible again. Previously proposed techniques avoid this problem by restricting hand motions to only those that are frontal to the camera [1] [3]-[5] [17]. To overcome this restriction, we propose to use multiple pre-calibrated cameras, so that parts invisible in one camera are still visible in at least another camera. While this is the right approach to the self-occlusion problem, more observations put more burdens on the already busy computer. This motivates further improvement of search efficiency. Although Rehg and Kanade [14] proposed to use multiple cameras for finger tracking, hand motion in this paper is much more complicated and we need to design and implement more efficient method.

Since it is infeasible to maintain dense sampling in high dimensional state spaces, two methods have been proposed to solve these problems. One is to reduce the state dimensionality and the other is to improve sampling and to make better prediction.

We reduced the dimensions of the hand motion space by PCA and further perform independent component analysis (ICA) to extract local features as ICA basis vectors [1].

To improve sampling efficiency, Rui et al. propose Unscented Particle Filter (UPF) [13]. The UPF uses the unscented Kalman filter to generate sophisticated proposal distributions that seamlessly integrate the current observation, thus greatly improving the tracking performance. This method needs to establish a system dynamics model. As for our 26-DOF problem, it is even hard to establish the system dynamics model. Deutscher et al. propose annealed particle filtering which is modified for searches in high dimensional state spaces [11]. It uses a continuation principle, based on annealing, to introduce the influence of narrow peaks in the fitness function, gradually. It is shown to be capable of recovering full articulated body motion efficiently. However, the experiment is done against a black background. Bray et al. propose smart particle filtering which combines the Stochastic Meta-Descent (SMD), based on gradient descent with particle filtering [17]. Their 3D hand tracking result is robust and accurate. However, they need depth

Manuscript received October 18, 2006.
Manuscript revised October 25, 2006.

maps generated by a structured light 3D sensor, which are not available in real time.

We propose to add prediction to particle filtering. Parameters in the next frame are predicted and more particles are accordingly generated for areas of higher likelihood. The method is straightforward but proven to very effective in significantly reducing search cost.

This paper is organized as follows: Section 2 describes hand model, and briefly shows how its dimension is reduced by ICA. Section 3 presents tracking by multiple cameras. Section 4 presents particle filtering with prediction. Section 5 presents experimental results. Conclusion is given in Section 6.

2. Hand Model and Dimension Reduction by ICA

In our study, a hand is rendered in OpenGL using spheres, cylinders, and rectangular parallelepiped. A hand can be described in this way: the base is a palm and five fingers are attached to the palm. Each finger has 4 DOF. 2 of 4 DOF correspond to the metacarpophalangeal joint (MP) and its abduction (ABD). The other 2 DOF correspond to the proximal interphalangeal joint (PIP) and the distal interphalangeal joint (DIP). It is shown in Fig. 1. Therefore, our hand model has 20 DOF. In addition, to represent the position and orientation of a hand, we need 6 more parameters, 3 for position and 3 for orientation. In total, the hand model has 26 DOF.

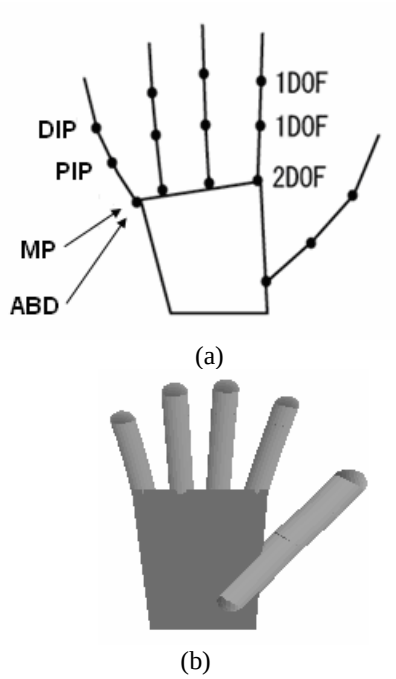

Figure 1. (a) Hand model with the name of each joint, and the degrees of freedom (DOF). (b) Hand model rendered in OpenGL.

We proposed an ICA-based representation of hand articulation [1]. It compresses the dimensionality of hand articulated motion very efficiently. Each ICA basis vector represents a finger motion. Thus the 20-DOF hand model can be represented by 5 DOF using this representation.

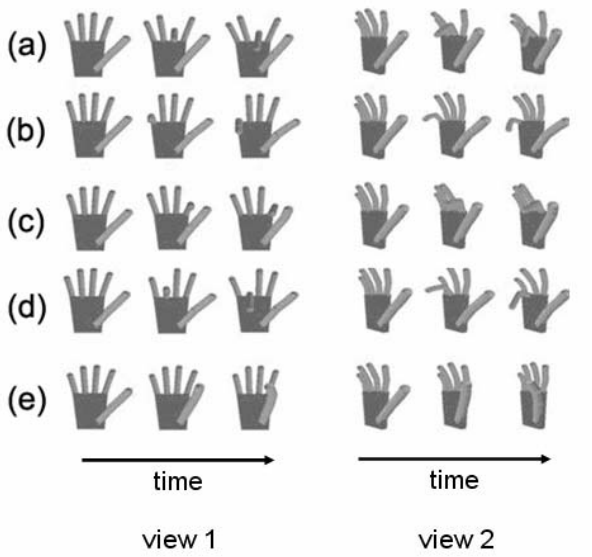

Figure 2. The ICA-based representation of hand articulation. (a)-(e) ICA basis motion, respectively.

Articulated hand motion is learned from the training data captured by a data glove. First, we perform PCA to reduce the dimensionality. Then we perform ICA to extract local finger motions. Each local finger motion corresponds to a particular finger motion as shown in Fig. 2.

The total dimension reduces to 11, with 5 for finger articulation, and 6 for global hand motion.

3. Tracking by Multiple Cameras

We perform camera calibration so that the intrinsic parameters and positions and orientations of the cameras recovered [15]. Once the cameras are calibrated, a hand model is projected onto the images, and the projected images are compared with real observations so that the parameters of the hand model can be estimated. Since calibrated cameras do not increase unknown parameters, more images do not mean more parameters. They merely bring more information.

In our currently experiments, we use two cameras looking at the hand, with the two cameras separated by roughly 90 degrees. This brings a great improvement over using a single camera, and is sufficient in handling occlusions.

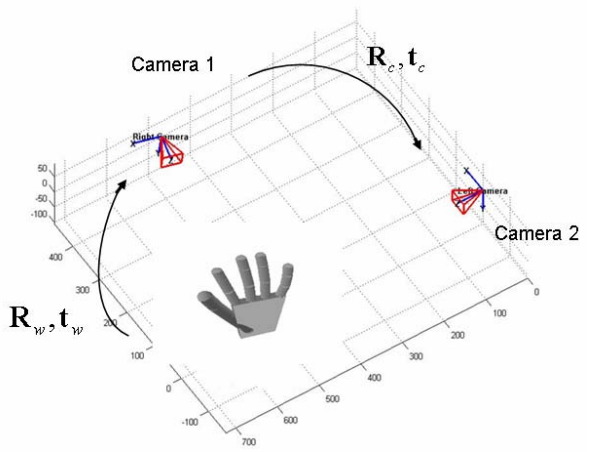

Figure 3. Relation between two cameras.

3.1 Relation Between the Hand Model and Two Cameras

The relation between two cameras is drawn as follows. The 3D coordinate system centered at optical center of camera 1 is $\mathbf{X}$. The 3D coordinate system centered at optical center of camera 2 is $\mathbf{X}'$. As depicted in Fig. 3, the rotation matrix and the translation vector from the coordinate system of camera 1 to the coordinate system of camera 2 are $\mathbf{R}_c, \mathbf{t}_c$. Then the relation between the two coordinate systems is given by

$$\mathbf{X} = \mathbf{R}_c \mathbf{X}' + \mathbf{t}_c . \quad (1)$$

The 3D coordinate system of hand model is $\mathbf{X}_m$. The rotation matrix and the translation vector from the coordinate system of hand model to the coordinate system of camera 1 are $\mathbf{R}_w, \mathbf{t}_w$. Then the relation between the two coordinate systems is given by

$$\mathbf{X}_m = \mathbf{R}_w \mathbf{X} + \mathbf{t}_w . \quad (2)$$

From (1) and (2), we can transform $\mathbf{X}_m$ to $\mathbf{X}$ and $\mathbf{X}'$, and then project the hand model onto the images.

3.2 Observation Model

We employ edge and silhouette information to evaluate the hypotheses. For edge information, we employ the Chamfer distance function [2]. First, we perform Canny edge detection to the input image. In the result image of edge detection, the edge pixels are black and other pixels are white. Then, at each pixel, we calculate the distance from each pixel to the closest edge point by using distance transformation. If the distance is over a threshold, the distance is set to the threshold. A distance map of the input image is obtained. Fig. 4 (b) shows an example of distance map. Then we project the edge of the hand model onto the distance map. We add all distances along the edge points of the projected hand model, and calculate the average of distances. Then the likelihood from the edge information is

$$p_{edge}(z_t \mid \mathbf{x}_t) \propto \exp\left[-\frac{(avaDist)^2}{2\sigma_{edge}^2} \right], \quad (3)$$

where avaDist is the average of distances.

In order to extract the silhouette of a hand region, we convert image color space from RGB to HSV (hue, saturation and brightness). Then the skin color region is extracted by using a threshold. Fig. 4 (c) shows an extracted silhouette. We calculate subtractions of the area of silhouette. The three calculated subtraction results are shown in Fig. 5. The subtractions of $a_I - a_O$ and

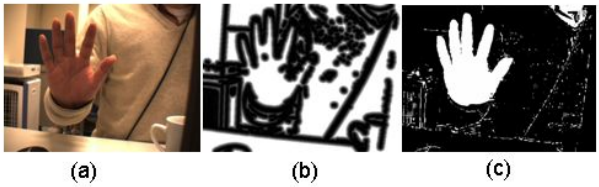

Figure 4. (a) Input image (b) Distance map of edge observation (c) Extracted silhouette

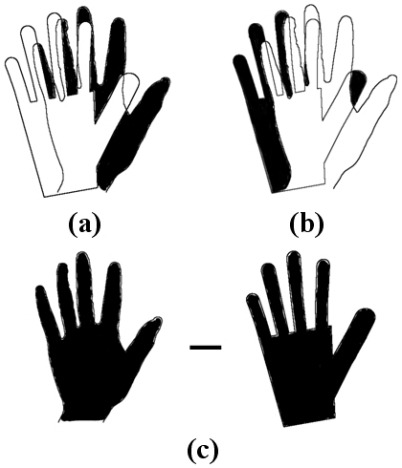

Figure 5. Areas of silhouette measurements. Black areas are the corresponding areas. (a) $a_I - a_O$, (b) $a_M - a_O$, (c) $a_I - a_M$. Note that a_I is the silhouette of input image, a_M is the silhouette of hand model, and a_O is the silhouette of overlap.

$a_M - a_O$ are used to measure the similarity of the hand position. The subtraction of $a_I - a_M$ is used to measure the similarity of hand finger pose. Then likelihoods from the silhouette information are

$$p_{silIO}(z_t \mid \mathbf{x}_t) \propto \exp\left[-\frac{(a_I - a_O)^2}{2\sigma_{silIO}^2}\right] \quad (4)$$

$$p_{silMO}(z_t \mid \mathbf{x}_t) \propto \exp\left[-\frac{(a_M - a_O)^2}{2\sigma_{silMO}^2}\right] \quad (5)$$

$$p_{silIM}(z_t \mid \mathbf{x}_t) \propto \exp\left[-\frac{(a_I - a_M)^2}{2\sigma_{silIM}^2}\right]. \quad (6)$$

Thus the final likelihood is

$$p(z_t \mid \mathbf{x}_t) \propto p_{edge}\, p_{silIO}\, p_{silMO}\, p_{silIM}. \quad (7)$$

When we use multiple cameras, the likelihood is

$$p(z_t \mid \mathbf{x}_t) \propto \prod_{i=1}^{n} p_i(z_t \mid \mathbf{x}_t), \quad (8)$$

where n is the number of cameras.

4. Particle Filtering with Prediction

4.1 Particle Filtering

The particle filtering algorithm [16] [18] is a sequential Monte Carlo method. The algorithm is powerful in approximating non-Gaussian probability distributions. Particle filtering is based on sequential importance sampling and Bayesian theory. With particle filtering, continuous distributions are approximated by discrete random sample sets, which are composed of weighted particles. The particles represent hypotheses of possible solutions and the weights represent likelihood.

There are three main steps in the algorithm: resampling, diffusion, and observation. The first step selects the particles for reproduction. In this step, particles that have heavier weights are more likely to be selected. Heavy-weight particles generate new ones, while light-weight particles are eliminated. The second step diffuses particles randomly. A part of space that is more likely to have a solution has more particles, while a part of space that is less likely to have a solution has fewer particles. The third step measures the weight of each particle

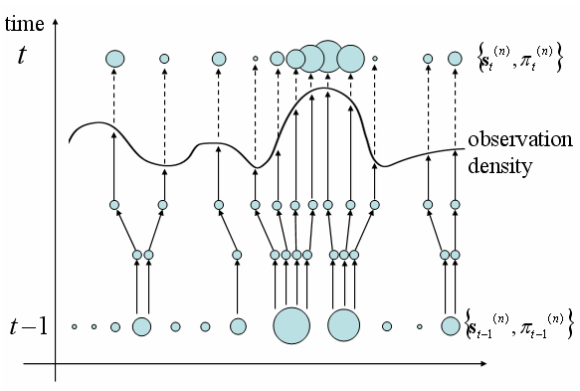

Figure 6. One time-step in particle filtering. There are 3 steps, resampling-diffusion-observation.

according to an observation density. Fig. 6 shows a pictorial description of particle filtering.

4.2 Particle Filtering with Prediction

The classical particle filtering requires an impractically large number of particles to follow rapid motions and to keep tracking correct. It becomes a serious problem when the tracking target has a high dimensional state space like hand tracking. In order to tackle this problem, we propose using prediction to generate better proposal distributions.

According to the Bayes rule, the hand pose of the current frame $\mathbf{x}_t$ can be estimated from the prior hand pose $\mathbf{x}_{t-1}$ as

$$p(\mathbf{x}_t \mid z_{1:t}) \propto p(z_t \mid \mathbf{x}_t)\, p(\mathbf{x}_t \mid z_{1:t-1}), \quad (9)$$

where

$$p(\mathbf{x}_t \mid z_{1:t-1}) = \int_{\mathbf{x}_{t-1}} p(\mathbf{x}_t \mid \mathbf{x}_{t-1})\, p(\mathbf{x}_{t-1} \mid z_{1:t-1}), \quad (10)$$

z_t is the observation of the current frame, $p(z_t \mid \mathbf{x}_t)$ is the likelihood distribution and $p(\mathbf{x}_t \mid \mathbf{x}_{t-1})$ is the transition probability distribution. (9) can be interpreted as the equivalent of the Bayes rule:

$$p(\mathbf{x} \mid z) \propto p(z \mid \mathbf{x})\, p(\mathbf{x}). \quad (11)$$

In particle filtering, the sequence of probability distributions is approximated by a large set of particles. Therefore, how to propagate the particles efficiently in areas of higher likelihood significantly affects tracking results. The particles are defined as follows: in order to represent a posteriori $p(\mathbf{x}_t \mid z_{1:t})$, we employ a time-stamped sample set, denoted $\{\mathbf{s}_t^{(n)}, n = 1, \cdots, N\}$. The sample set is weighted by the observation density

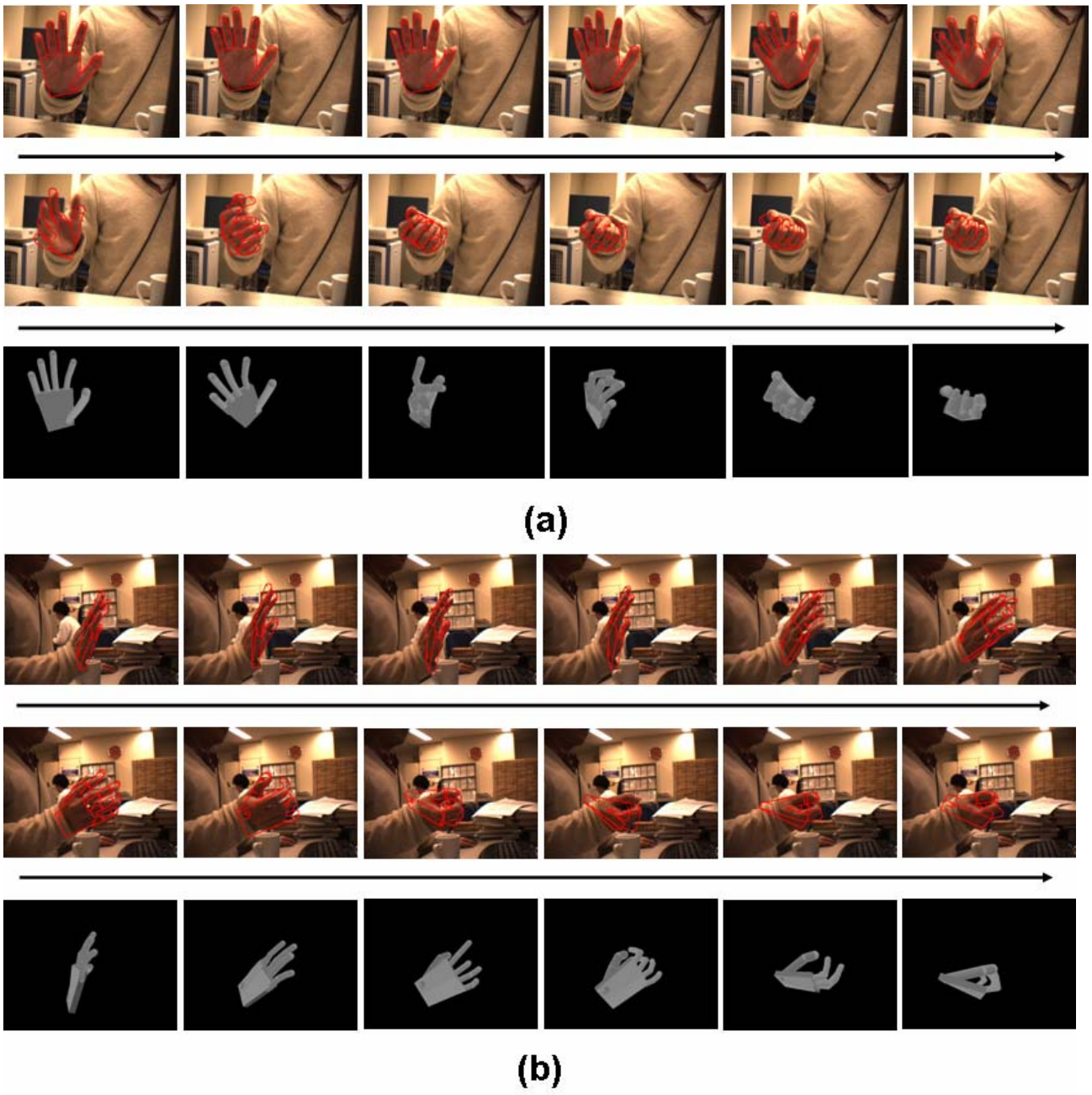

Figure 7. Tracking result by two cameras. (a) Camera 1 view. (b) Camera 2 view. The projection of hand model's edge is drawn on the images by red lines. The CG models are examples of some corresponding hand models.

$\pi_t^{(n)} = p(z_t \mid \mathbf{x}_t = \mathbf{s}_t^{(n)})$, where the weights $\pi_t^{(n)}$ are normalized so that $\sum_N \pi_t^{(n)} = 1$. Then the sample set $\{\mathbf{s}_t^{(n)}, \pi_t^{(n)}\}$ represents the posteriori $p(\mathbf{x}_t \mid z_{1:t})$. The sample set of the posteriori is propagated from $\{\mathbf{s}_{t-1}^{(n)}, \pi_{t-1}^{(n)}\}$ which represents $p(\mathbf{x}_{t-1} \mid z_{1:t-1})$ as shown in Fig. 6. The transition probability distribution $p(\mathbf{x}_t \mid \mathbf{x}_{t-1})$ affects $p(\mathbf{x}_t \mid z_{1:t-1})$, which in turn affects $p(\mathbf{x}_t \mid z_{1:t})$.

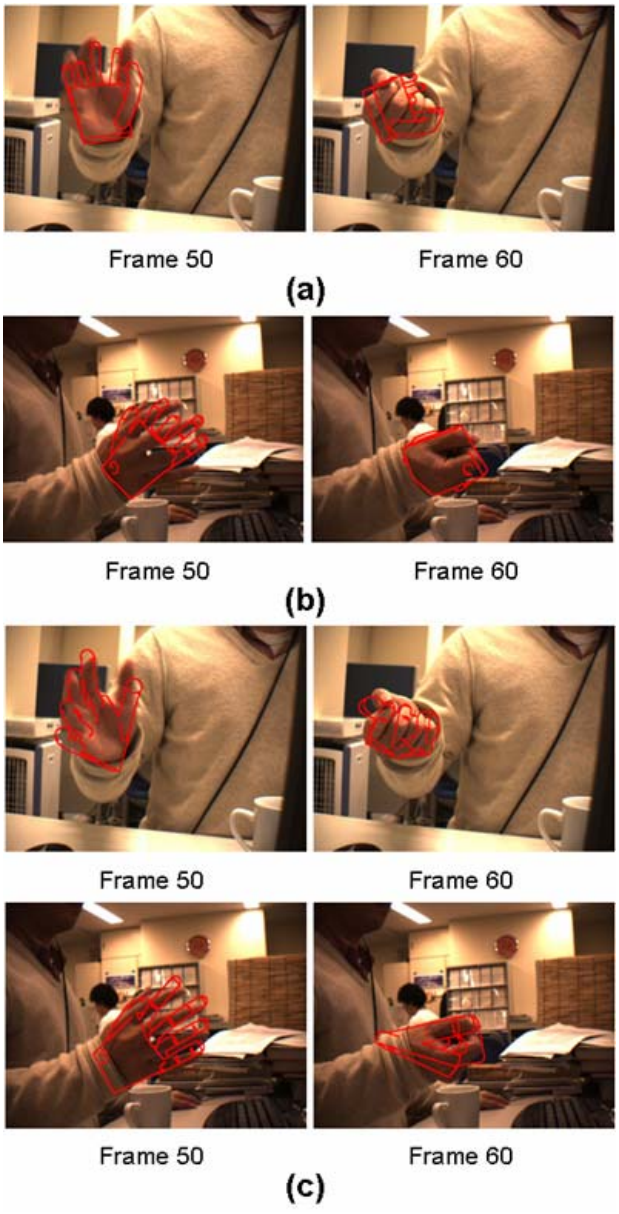

Figure 8. (a) Result 1 by single camera. (b) Result 2 by single camera. (c) Result by two cameras.

In particle filtering, $p(\mathbf{x}_t \mid \mathbf{x}_{t-1})$ is modeled by a dynamical model. The simplest dynamical model [1] [4]-[6] [11] [17] is

$$\mathbf{s}_t^{(n)} = \mathbf{s}_{t-1}^{(n)} + \mathbf{B}, \qquad (12)$$

where $\mathbf{B}$ is a multivariate Gaussian distribution with covariance $\mathbf{P}$ and mean $\mathbf{0}$. However, this simple dynamical model does not propagate the particles efficiently and many particles are wasted in areas of lower likelihood.

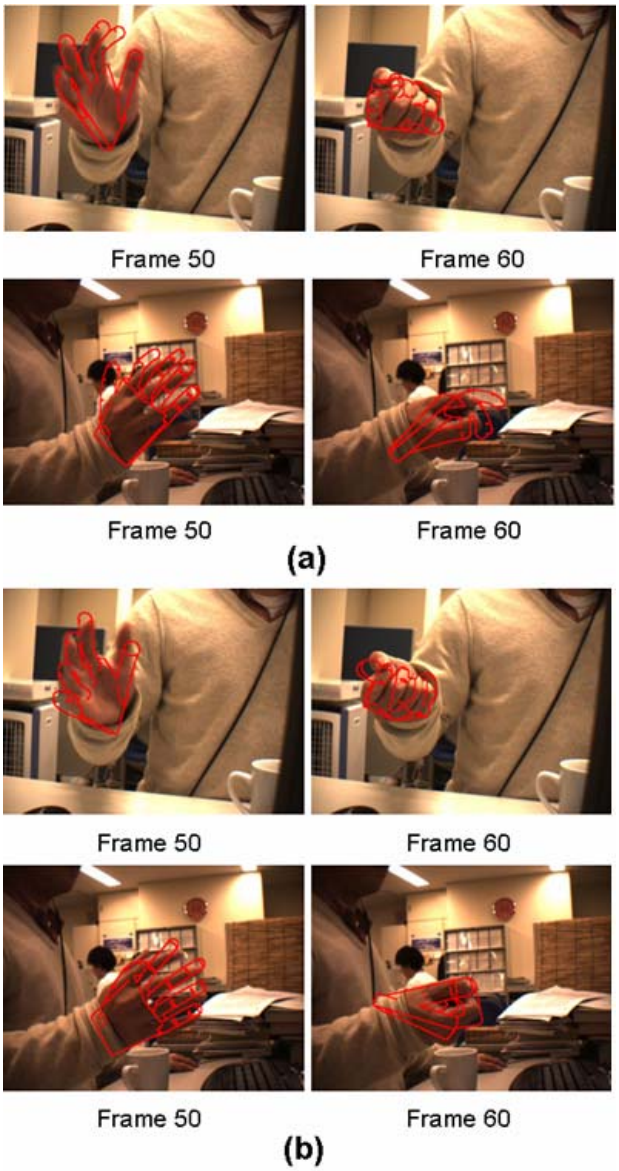

Figure 9. (a) Tracking result without prediction. (b) Tracking result with prediction.

To overcome these difficulties, we simply use the first-order approximation of Taylor series expansion for prediction:

$$\mathbf{s}_t^{(n)} = \mathbf{s}_{t-1}^{(n)} + \frac{\partial \mathbf{s}_{t-1}^{(n)}}{\partial t}\Delta t + \mathbf{B}. \qquad (13)$$

We also tried to use the second-order approximation of Taylor series expansion

$$\mathbf{s}_t^{(n)} = \mathbf{s}_{t-1}^{(n)} + \frac{\partial \mathbf{s}_{t-1}^{(n)}}{\partial t}\Delta t + \frac{1}{2}\frac{\partial^2 \mathbf{s}_{t-1}^{(n)}}{\partial t^2}\Delta t^2 + \mathbf{B}. \qquad (14)$$

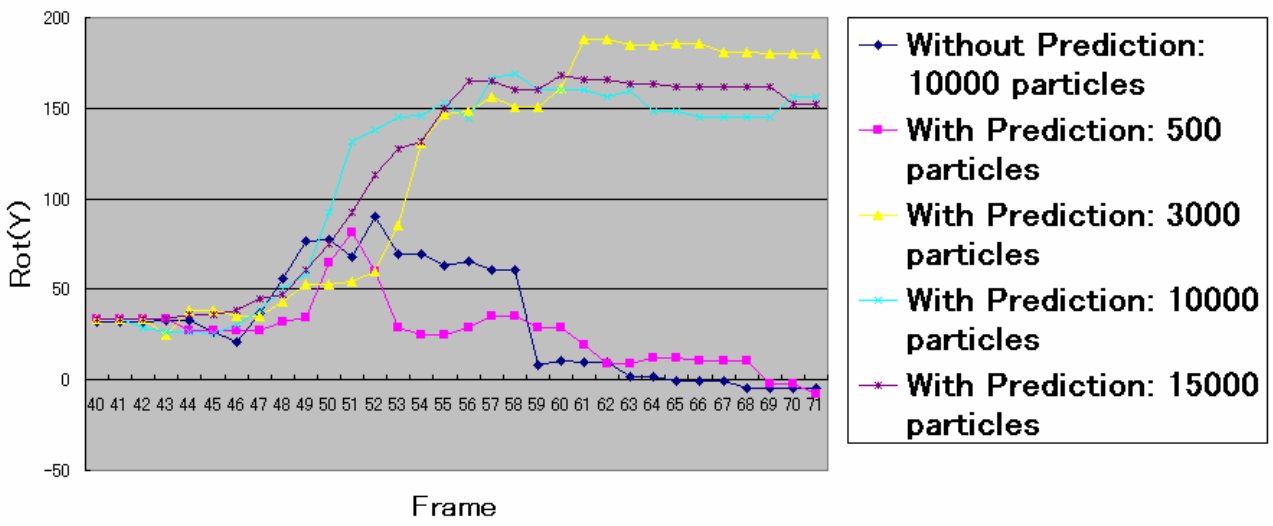

Figure 10. Trajectory of the rotation around Y axis (unit: degrees).

However, the tracking gets trapped in local minima. The reason is that the second derivative cannot be estimated accurately due to noise.

5. Experimental Results

The performance of our method was tested by using real image sequences. A movie (avi) file of the results is available on the web at http://www.cvg.is.ritsumei.ac.jp/~kmakoto/.

We manually initialize the hand model to match roughly with the hand at the first frame. Then the algorithm automatically tracks the hand while it moves freely.

5.1 Tracking Rapid Motions against a Cluttered Background

Fig. 7 shows the tracking result of our method. The sequences include rapid motion, large rotations angle against a camera, occlusions and a cluttered background. The experiment was run using 10000 particles per frame. The tracking correctly estimated hand position and motion throughout the sequence. In the following subsections, we compare our method to the method by a single camera, and the method without prediction.

5.2 Tracking by a Single Camera

If we use only a single camera, the tracking result becomes Fig. 8 (a) and (b). We tried the experiment by a single camera to two image sequences respectively. Fig. 8 (c) is the result by two cameras. In Fig. 8 (a), at frame 50, the hand orientation is slightly incorrect and then the error becomes larger, finally, at frame 60, the hand orientation is completely incorrect. In Fig. 8 (b), at frame 50, the hand orientation is incorrect and then the error becomes larger, finally at frame 60, the tracking estimated that the hand fingers exist at the hand wrist position. From the results, we can see that occlusion is a severe problem for tracking by a single camera but is not a problem for multiple cameras.

5.3 Tracking Without Prediction

In the next experiment, we compared the methods with prediction and without prediction. Fig. 9 (a) is the result without prediction and Fig. 9 (b) is that with prediction. In Fig. 9 (a), at frame 50, the hand orientation is slightly incorrect, and then the error becomes larger and finally, at frame 60, the tracking estimated that the hand is upside down comparing with the real hand.

5.4 The Number of Particles

We also did experiments with different numbers of particles per frame in order to watch how many particles are suitable for the tracking. We show the trajectory of the rotation around Y axis in Fig. 10.

We did the experiment with 500 particles, 3000 particles, 10000 particles and 15000 particles. The results have dramatic change when we increase the number of particles from 500 to 10000. And the results only have slight change when we increase the number of particles from 10000 to 15000. Therefore, 10000 is the optimized number of particles for this hand motion.

6. Conclusions

In this paper we proposed an articulated hand motion tracking by multiple cameras. This method is useful for gesture recognition. Tracking a free hand motion against a cluttered background was unachievable in previous methods because hand fingers are self-occluding. To improve search efficiency, we proposed adding prediction to particle filtering so that more particles are generated in areas of higher likelihood. The experimental results show that our method can correctly and efficiently track the hand motion throughout the image sequences even if hand motion has large rotation against a camera.

References

[1]M. Kato, Y. W. Chen, and G. Xu, "Articulated Hand Tracking by PCA-ICA approach," IEEE Proc. FG06, pp. 329-334, 2006.

[2] B. Stenger, A. Thayananthan, P. H. S. Torr, and R. Cipolla, "Filtering using a tree-based estimator," Proc. ICCV03, pp.1102-1109, Nice, France, October 2003.

[3] H. Zhou and T. S. Huang, "Tracking Articulated Hand Motion with Eigen Dynamics Analysis," Proc. ICCV03, pp.1102-1109, Nice, France, October 2003.

[4] Y. Wu, John Lin, and T. S. Huang, "Capturing natural hand articulation," Proc. ICCV01, pp.426-432, Vancouver, July 2001.

[5] Y. Wu, John Lin, and T. S. Huang, "Analyzing and Capturing Articulated Hand Motion in Image Sequences," IEEE Trans. PAMI, Vol. 27, No. 12, December 2005.

[6] W. Y. Chang, C. S. Chen, and Y. P. Hung, "Appearance-Guided Particle Filtering for Articulated Hand Tracking," Proc. CVPR05, Vol. 1, pp.235-242, June 2005.

[7] J. Lee and T. Kunii, "Model-based analysis of hand posture," IEEE Computer Graphics and Applications, 15: pp.77-86, Sept. 1995.

[8] M. Isard and A. Blake, "Contour tracking by stochastic propagation of conditional density," Proc. of European Conf. on Computer Vision, pp. 343-356, Cambridge, UK, 1996.

[9] M. Isard and A. Blake, "CONDENSATION-conditional density propagation for visual tracking," Int. J .Computer Vision, 1998.

[10] A. Blake and M. Isard, Active Contours. Springer, London, 1998.

[11] J. Deutscher, A. Blake, and I. Reid, "Articulated Body Motion Capture by Annealed Particle Filtering," Proc. CVPR00, Vol. 2, pp. 126-133, 2000.

[12] C. Sminchisescu and B. Triggs, "Covariance Scaled Sampling for Monocular 3D Body Tracking," Proc. CVPR01, Vol. 1, pp. 447-454, 2001.

[13] Y. Rui and Y. Chen, "Better Proposal Distributions: Object Tracking Using Unscented Particle Filter," Proc. CVPR01, Vol. 2, pp. 786-793, 2001.

[14] J. Rehg and T. Kanade, "Model-Based Tracking of Self Occluding Articulated Objects," Proc. ICCV95, pp. 612-617, 1995.

[15] J. Y. Bouguet, MRL-Intel Corp. "Camera Calibration Toolbox for Matlab," available at http://www.vision.caltech.edu/bouguetj/calib_doc/

[16] P. M. Djuric, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Miguez, "Particle filtering," IEEE Signal Processing Magazine, pp.19-38, Sept. 2003..

[17] M. Bray, E. K. Meier, and L. V. Gool, "Smart Particle Filtering for 3D Hand Tracking," IEEE Proc. FG04, pp. 675-680, 2004.

[18] M. Sanjeev Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking," IEEE Trans. Signal Processing, Vol. 50, No. 2, pp. 174-188, Feb 2002.

Makoto Kato received the B.S. and M.S. degrees in Computer Science from Ritsumeikan University in 2001 and 2003, respectively. He is currently a Ph.D. student in Ritsumeikan University. His research interests include statistical pattern recognition, computer vision, and machine learning.

Gang Xu is a professor of computer science, Ritsumeikan University. He received his Ph.D. degree from Osaka University in 1989, and since then he has held teaching and research positions in Osaka University, Ritsumeikan University, Harvard University, Microsoft Research Asia and Motorola Australian Research Centre. He also founded 3D MEDiA Company Limited in 2000, specialized in 3-dimensional image processing and photogrammetry. He authored and co-authored 3 books in computer vision.