FINGERS IMAGE TRACKING ON OMNIDIRECTION CAMERA BY CONDENSATION ALGORITHM

Norikazu Ikoma
Faculty of Engineering,
Kyushu Institute of Technology,
1-1 Sensui-cho, Tobata-ku,
Kita-Kyushu, Fukuoka, Japan
email: ikoma@comp.kyutech.ac.jp

Masahide Sakata
Faculty of Engineering,
Kyushu Institute of Technology,
1-1 Sensui-cho, Tobata-ku,
Kita-Kyushu, Fukuoka, Japan
email: sakata@sys2.comp.kyutech.ac.jp

Motonori Doi
Dept. of Telecom. and Comp. Net.,
Osaka Electro-Communication Univ.
18-8 Hatsu-cho,
Neyagawa, Osaka, Japan
email: doi@isc.osakac.ac.jp

ABSTRACT

Tracking of fingers image on omnidirection camera has been investigated which aims at developing of human friendly interface using fingers' gesture. Where the way of using the omnidirection camera is different from a normal usage in which the user holds the camera with his/her fingers and move them. The interface can use the motion of the fingers to establish a communication between the user and computer system in human friendly way. To achieve this, it is necessary for the system to track the fingers in the dynamic image of omnidirection camera. We have employed CONDENSATION algorithm for the tracking. The algorithm uses pre-defined shape of finger called template, and estimates the parameters of affine transformation to adjust the template to the image. The estimation is performed with many number of particles where each particle has an realization of the affine parameters. Applying three steps called prediction, observation, and selection, we have updated particles which approximates a posteriori distribution of the affine parameters given the image sequence up to current time. Real image experiments show the performance of the tracking method.

KEY WORDS

Omnidirection camera, dynamic image, finger image, tracking, CONDENSATION algorithm.

1 Introduction

Recent technologies allow us to use various small sensors for human-machine interfaces. Vision sensor is one of the most promising ways of sensing since it is possible to use natural gesture of human to communicate with the computer system through the image. There are many interesting topics to be investigated with vision sensor, and the related field of computer vision is still challenging. Among them, we focus on omnidirection camera and its special use as follows.

There is a research that uses omnidirection camera as a new human-machine interface with finger gesture [1]. It proposes to use the omnidirection camera in different way, i.e., holding it with fingers as shown in figure 1. Figure 2 shows typical image frames obtained by the omnidirection camera using in this way.

To achieve the interface using the fingers' image of omnidirection camera, it is necessary to know the state of the fingers, such as position, angle, and their motion speeds. For this purpose, we propose to use CONDENSATION algorithm [4], which is suit for tracking the object with known shape. In the CONDENSATION algorithm, the typical shape is pre-defined and its affine transformation is applied to the image. Thus we estimate the affine parameter to know the object's shape on the image. For this estimation, the CONDENSATION algorithm effectively uses many realizations of the parameters, which are called 'samples' or 'particles'.

In the rest of this paper, we formulate the problem in mathematical model for fingers tracking where the number of fingers is assumed to the known and fixed. After explaining the CONDENSATION algorithm, we will show the results of experiments with fingers 1, 2, and 3.

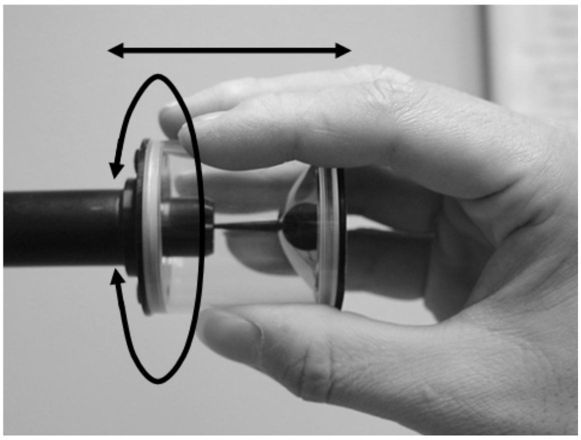

Figure 1. A new way to use omnidirection camera with fingers aimed at human friendly interface; gripping the camera with fingers. There are typical ways of motions; rotation of fingers (denoted by circulated arrow), sliding the fingers (denoted by line arrow), motion of each finger and tapping of a finger.

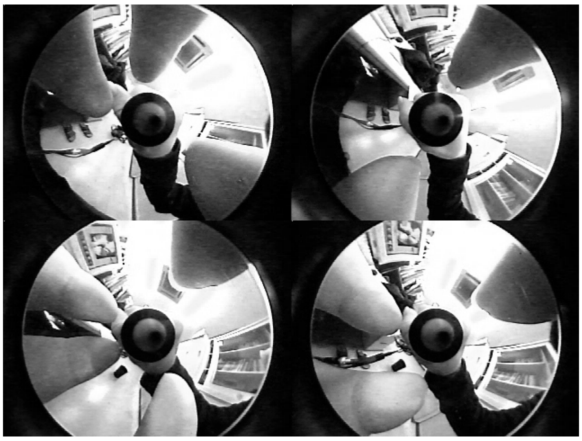

Figure 2. Typical image frames obtained by omnidirection camera with fingers. Upper-left:begin with three fingers, upper-right:changed to two fingers, lower-left:four fingers, and lower-right:three fingers again. Tapping of finger is also a typical motion of fingers.

2 Model

2.1 Finger state

Figure 3 illustrates how a finger reflects to the omnidirection camera image. We model the finger image based on this figure. That is, fingers are assumed to have typical shape as shown in the figure, and each finger reflecting to the image is identified by its position θ, width w, length h, and inclination ϕ. Collection of these parameters form a state vector of a finger

$$\mathbf{x} = [\theta, w, h, \phi]\,. \tag{1}$$

We put index of finger for the state vector such as $\mathbf{x}_f$ with $f = 1, 2, \cdots, F$ with F be the number of fingers (fixed and known in this paper). We also put discrete time index $k = 1, 2, \cdots$ for each state vector $\mathbf{x}_f$, such as $\mathbf{x}_{f,k}$, to represent time evolution of the state.

For the time evolution of the fingers, we are supposed to have few knowledge, i.e., fingers are smoothly moving and motion of fingers are mutually independent. Then the model for the time evolution is written in a stochastic difference equation

$$\mathbf{x}_{f,k} = \mathbf{x}_{f,k-1} + \mathbf{v}_{f,k} \tag{2}$$

where $\mathbf{v}_{f,k}$ is random vector called 'system noise', which represents our lack of knowledge for the evolution. Typically, Gaussian distribution with zero mean and diagonal covariance is assumed to the system noise term. Alternatively, we can represent the time evolution in a way of conditional density form

$$\mathbf{x}_{f,k} \sim f(\cdot|\mathbf{x}_{f,k-1}). \tag{3}$$

Then, all fingers' state is represented by a collection of the state vectors

$$\mathbf{X}_k = [\mathbf{x}_{1,k}, \mathbf{x}_{2,k}, \cdots, \mathbf{x}_{F,k}]'. \tag{4}$$

The time evolution for all fingers is also represented in a form of stochastic difference equation

$$\mathbf{X}_k = \mathbf{X}_{k-1} + \mathbf{V}_k \tag{5}$$

with $\mathbf{V}_k = \left[\mathbf{v}'_{1,k}\mathbf{v}'_{2,k}, \cdots, \mathbf{v}'_{F,k}\right]$, and in a form of conditional density

$$\mathbf{X}_k \sim f(\cdot|\mathbf{X}_{k-1}). \tag{6}$$

with $f(\mathbf{X}_k|\mathbf{X}_{k-1}) = \prod_{f=1}^{F} f(\mathbf{x}_{f,k}|\mathbf{x}_{f,k-1})$.

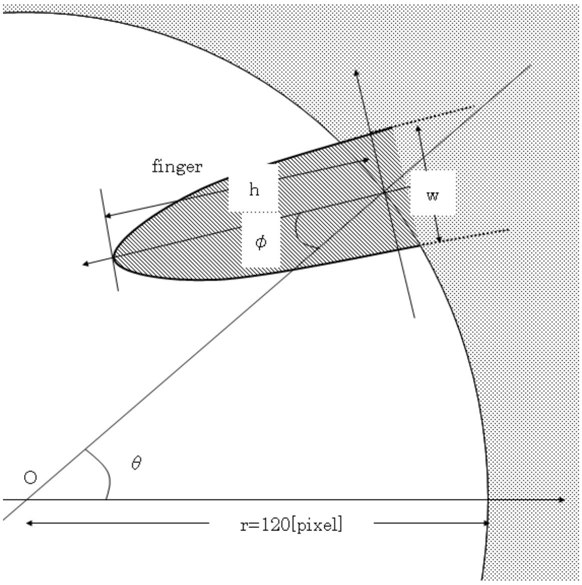

Figure 3. Modeling of finger in omnidirection camera image. 'O' is the center of omnidirection camera image.

2.2 Likelihood

Typical shape of tracking object is assumed and it is represented by a template in CONDENSATION algorithm [4]. In our method for fingers tracking, the template for finger is defined as shown in figure 4. Where $P = 13$ control points are place at certain positions of the outline of the finger, and line segments to measure the image are placed on each control point.

Let $\mathbf{I}_k$ be the image at time k. For a given finger state at time k, $\mathbf{X}_k$, we translate the template according to each finger state $\mathbf{x}_{f,k}$ ($f = 1, 2, \cdots, F$) and put them on the image $\mathbf{I}_k$. Then we obtain intensity functions of the image along with the line segments of the translated templates, such that for j-th control point of f-th finger,

$$I_{j,k}(z; \mathbf{x}_{f,k}) \tag{7}$$

In (7), z represents the position on the line segment of j-th control point with its origin be at the position of the control point and the axis be directed to the outward of the finger shape of the template.

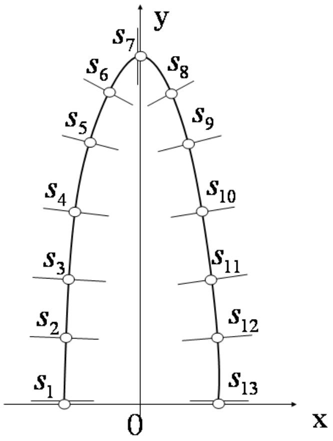

Figure 4. Template of finger used in CONDENSATION algorithm. Curve denotes the outline of finger, circles denote the control points, and each line crossing the circle denotes the line segment. There are $P = 13$ control points.

By taking 1st difference of the intensity function (7) and extracting the large difference position nearest to the origin, we have observation of the outline of the finger along with the line segments. We denote them by

$$\mathbf{z}_{f,k} = [z_{1,f,k}, z_{2,f,k}, \cdots, z_{P,f,k}] \tag{8}$$

for f-th finger at time k. Note that (13) is actually depending on the state $\mathbf{x}_{f,k}$ since it uses eq.(7).

Observation model is represented by observation density $p(z)$ along with each line segment. Then, likelihood function of f-th finger is derived from the observation model as

$$h(\mathbf{z}_{f,k}|\mathbf{x}_{f,k}) \equiv \prod_{j=1}^{P} p(z_{j,f,k}) \tag{9}$$

under mutually independence assumptions for the observation densities of line segments. Typically for the observation density $p(z)$, we use symmetrically trimmed version of Gaussian distribution such as

$$p(z;\sigma^2,l) \propto \sigma^{-1} \exp\left\{- \min\left(z^2,l^2\right)/2\sigma^2\right\} \tag{10}$$

with σ^2 be variance, and $2l$ be the maximum lenght of the line segment.

Likelihood function for all fingers are the product of that of each finger under mutually independence assumption

$$h(\mathbf{I}_k|\mathbf{X}_k) \equiv h(\mathbf{Z}_k|\mathbf{X}_k) = \prod_{f=1}^{F} h(\mathbf{z}_{f,k}|\mathbf{x}_{f,k}) \tag{11}$$

with $\mathbf{Z}_k = [\mathbf{z}_{1,k}, \mathbf{z}_{2,k}, \cdots, \mathbf{z}_{F,k}]$. Note that the likelihood (11) is highly complex due to the nonlinearlity involved in the observation scheme.

3 Estimation

The objective is to estimate the fingers' state based on the image sequence. It is performed in a framework of state estimation with two densities in previous section, that is, (6) be the system density and (11) be the observation density. The task of state estimation is to obtain a posteriori density $p(\mathbf{X}_k|\mathbf{I}_{1:k})$ where we have used a special notation $\mathbf{I}_{1:k} \equiv (\mathbf{I}_1, \mathbf{I}_2, \cdots \mathbf{I}_k)$.

We need to use an approximation method to achieve this task since it has no closed-form solution due to the complex likelihood function. As the approximation method, we have employed CONDENSATION algorithm [4]. It is a special case of particle filters, which is a special class of Sequential Monte Carlo method [2] into the context of optimal filtering. Similar algorithms have been proposed in different names, such as Bootstrap filter [3], and Monte Carlo filter [5].

The key idea of these algorithms is to approximate the posterior by many samples according to the distribution. When the new observation becomes available, the algorithm updates the set of particles based on three steps; prediction, observation, and selection.

3.1 Initialization

Let the initial particles be $\left\{\mathbf{X}_0^{(i)}\right\}_{i=1}^{M}$, where M is the number of particles. The initial particles approximates the given initial distribution $p(\mathbf{X}_0|\mathbf{I}_{0:0})$.

3.2 Iteration

Starting from the initial particles and letting $k := 1$, the update proceeds with three steps below, followed by time increment $k := k + 1$;

3.2.1 Prediction Step

Firstly, in the prediction step, we draw particles at current time k by

$$\tilde{\mathbf{X}}_k^{(i)} \sim f(\cdot|\mathbf{X}_{k-1}^{(i)}) \tag{12}$$

for $i = 1, 2, \cdots, M$. Note that $\left\{\tilde{\mathbf{X}}_k^{(i)}\right\}_{i=1}^{M}$ approximates one-step-ahead prediction distribution $p(\mathbf{X}_k|\mathbf{I}_{1:k-1})$.

3.2.2 Observation Step

Next, in the observation step, calculate likelihood value for each particle

$$\alpha_k^{(i)} \propto h(\tilde{\mathbf{X}}_k^{(i)}|\mathbf{I}_k) \tag{13}$$

with normalization such that $\sum_{i=1}^{M} \alpha_k^{(i)} = 1$. Calculated (13) for $i = 1, 2, \cdots, M$. Then we obtain particles weighted by the likelihood, $\left\{ \left(\tilde{\mathbf{X}}_k^{(i)}, \alpha_k^{(i)} \right) \right\}_{i=1}^{M}$ which approximate the posterior distribution $p(\mathbf{X}_k | \mathbf{I}_{1:k})$.

3.2.3 Selection Step

Finally, in the selection step, a procedure so-called resampling is done, which is M repetition of sampling with replacement with probability $\alpha_k^{(i)}$ for particle $\tilde{\mathbf{X}}_k^{(i)}$. It can be formally written by

$$\mathbf{X}_k^{(i)} = \begin{cases} \tilde{\mathbf{X}}_k^{(1)} & \text{with prob.} \quad \alpha_k^{(1)} \\ \tilde{\mathbf{X}}_k^{(2)} & \text{with prob.} \quad \alpha_k^{(2)} \\ \vdots & \qquad\qquad\quad \vdots \\ \tilde{\mathbf{X}}_k^{(M)} & \text{with prob.} \quad \alpha_k^{(M)} \end{cases} \tag{14}$$

Then we have a new set of M particles $\left\{ \mathbf{X}_k^{(i)} \right\}_{i=1}^{M}$ which also approximates the posterior distribution $p(\mathbf{X}_k | \mathbf{I}_{1:k})$.

3.3 Calculation of the estimate

According to the algorithm above, we obtain a set of particles which approximate the posterior distribution. To use the result in engineering purpose including the human interface that we are considering, we need to calculate the estimate value from the approximated distribution. For this, we employ the average of the distribution such that

$$\bar{\mathbf{X}}_k = \frac{1}{M} \sum_{i=1}^{M} \mathbf{X}_k^{(i)}. \tag{15}$$

4 Experiments

We have conducted three experiments with fingers $F = 1$, 2, and 3. For all experiment, conditions for estimation are as follows; the number of particles is $M = 1,000$, observation noise variance is $\sigma^2 = 25$, with trimming length $l = 3\sigma$. For system noises, mutually independent Gaussian distributions with variances $\tau_\theta^2 = 5.0$, $\tau_w^2 = 1.0$, $\tau_h^2 = 1.0$, and $\tau_\phi^2 = 2.0$ are used. The initial distributions are as follows; For θ, $N(\bar{\theta}, 0, 5)$ with $\bar{\theta}$ be obtained manually from the initial image is used. For w and h, uniform distributions $U[w_{min}, w_{max}]$ and $U[h_{min}, h_{max}]$ with $w_{min} = 10$,$w_{max} = 30$ and $h_{min} = 20$,$h_{max} = 80$ are used. For ϕ, $N(0, \mu_\phi^2)$ with $\mu_\phi^2 = 3.0$ is used.

Results are shown in figure 5, 6, and 7. In each figure, control points and template, which are calculated based on eq.(15), are written over the original image for specific frames (shown in captures). Looking at the figures, we can see that mostly correct trackings are obtained.

5 Conclusion

We have proposed a tracking model of fingers in omnidirection camera image, which is a special use of omnidirection camera holding it with fingers aiming at a new human friendly user interface using fingers image. The model is currently for fixed and known number of fingers. Four variables, i.e., position, width, length, and inclination of finger are the parameters to be estimated for one finger. For the estimation of the parameters, we have proposed to use CONDENSATION algorithm [4], which is a special case of sequential Monte Carlo [2], or it is called 'particle filters'. The experimental results shows the performance of the proposed method.

In future works, we will extend the method available for variable number of fingers. Incorporation of the nonlinear equation of the mirror of the omnidirection camera is necessary to achieve more accurate performance of the tracking. On line implementation of the algorithm is also important for our purpose, which is to develop a new human friendly interface using the finger images.

Acknowledgements

This study was carried out under the ISM Cooperative Research Program (2004-ISM-CRP-2020).

References

[1] M.Doi, S.Ueda, and K.Akiyama, "Human Interface based on Finger Gesture Recognition using Omni-Directional Image Sensor", *Proc. of the 2003 IEEE Int. Sympo. on Virtual Environments, Human-Computer Interfaces, and Measurement Systems (VECIMS 2003)*, Lugano, Switzerland, 2003, 68-72.

[2] A.Doucet, J.F.G.de Freitas, and N.J.Gordon (eds): *Sequential Monte Carlo Methods in Practice*, (New York, Springer, 2001).

[3] N.J.Gordon, D.J.Salmond, and A.F.M.Smith: "Novel approach to nonlinear / non-Gaussian Bayesian State Estimation", *IEE Proceedings-F*, 140(2), 1993, 107-113.

[4] M.Isard and A.Blake, "CONDENSATION – Conditional Density Propagation for Visual Tracking", *Int. J. of Computer Vision*, 29(1), 1998, 5-28.

[5] G.Kitagawa: "Monte Carlo filter and smoother for non-Gaussian nonlinear state space models", *J. of Computational and Graphical Statistics*, 5(1), 1996, 1-25.

[6] M.Sakata, N.Ikoma, and M.Doi, "Tracking of fingers in dynamic image of omnidirection camera by CONDENSATION algorithm", *Abst. of the 36th ISCIE International Symposium on Stochastic System Theory and its Applications*, Saitama, Japan, 2004, 3-4. (Proceedings will be published)

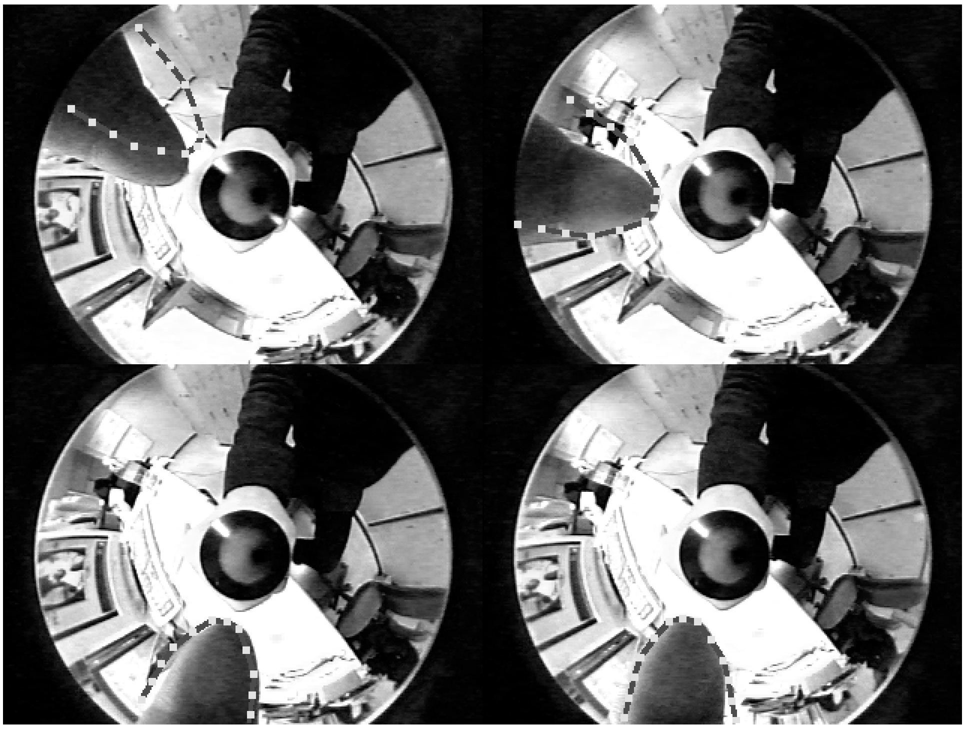

Figure 5. Tracking result for one finger case. Upper left: 1st frame, upper right: 26th frame, lower left: 51st frame, and lower right: 76th frame.

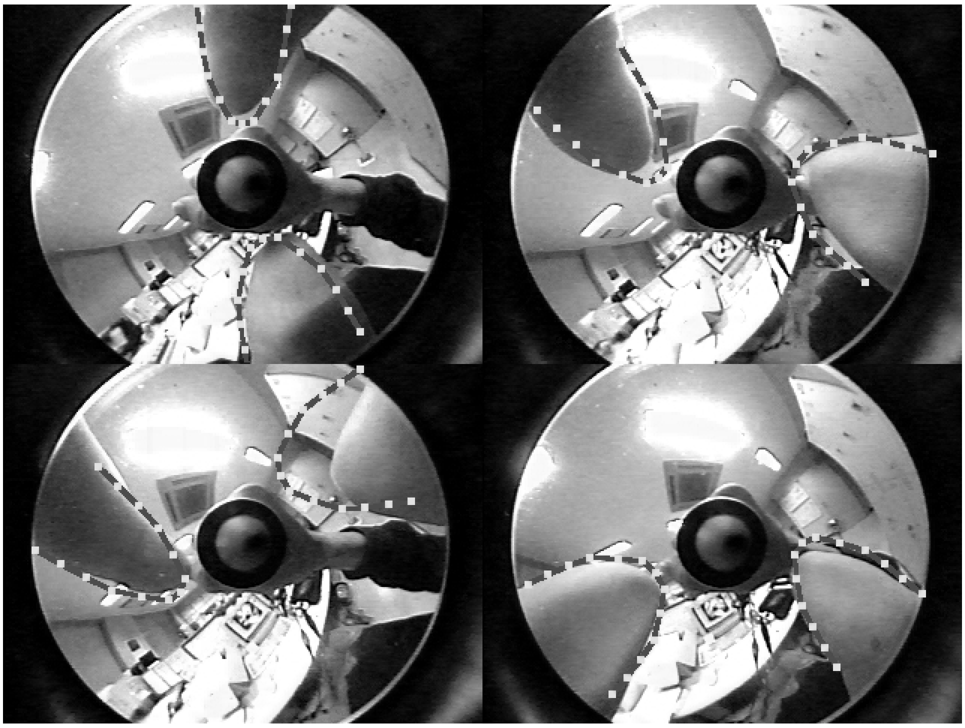

Figure 6. Tracking result for two fingers case. Upper left: 1st frame, upper right: 26th frame, lower left: 51st frame, and lower right: 76th frame.

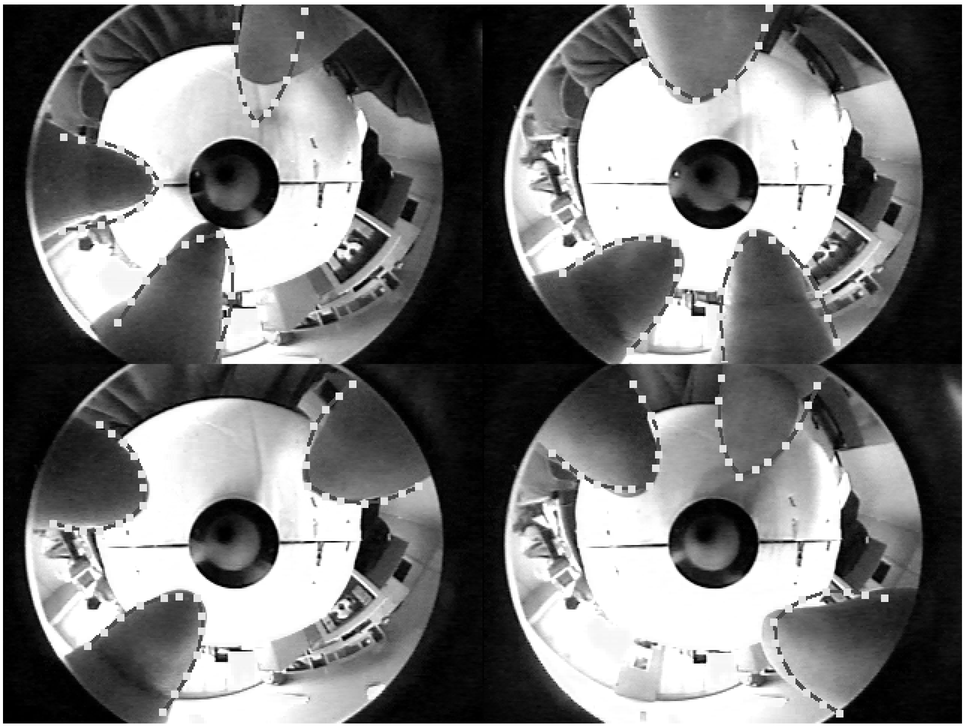

Figure 7. Tracking result for three fingers case. Upper left: 1st frame, upper right: 26th frame, lower left: 51st frame, and lower right: 76th frame.