The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

An Object-Tracking Algorithm Based on Particle Filtering with Region-Based Level Set Method

Xianfeng Fei and Koichi Hashimoto

Abstract— **In this paper, to realize robust tracking, we propose a particle filter (PF) model to track a single paramecium. The proposed PF model consists of a system dynamical model and an observation model. The information about our tracking object is described by a state vector and the system state is assumed to evolve according to the system dynamical model. The parallel region-based level set method with displacement correction (PR-LSM-DC) proposed in our previous work now works as the measurements for the PF model. The tracking is achieved by estimating the state of a moving object from the observations. Experiments show that with motion prediction using the PF model, we increase the robustness of tracking and extend the duration of single paramecium tracking. The 2 [ms] computational time indicates that we developed an algorithm and a computer aided system which achieves nonrigid single micro-organisms tracking in real-time as they deform, move and collide with others under optical microscope.**

I. INTRODUCTION

Visual single object tracking is an important task for many applications such as video surveillance system, mobile robots, medical diagnosis, microorganism observation and so on. Visual single object tracking still remains a challenging problem, because the tracking objects are always subject to deformation, various illuminations and collisions. In general, visual target tracking algorithms can be divided into two categories: deterministic methods and stochastic methods. The deterministic method is usually formulated as an optimization problem solved by minimizing the energy function. The stochastic approach is represented by state-space models, and the tracking problem is treated as a state-estimation problem.

In our previous work, we have been using parallel region-based level set method with displacement correction (PR-LSM-DC) which is one of deterministic methods to track a single paramecium [1], [2], [3], [4]. A parallel region-based level set method (PR-LSM) is used to detect the boundary of tracking object which is for calculating the centroid of object. However, when collisions happen, the detected contour of the object spreads to the other obstacle which induces target missing and tracking failure. Therefore, after the collision is detected, we correct the boundary detection result of PR-LSM by translating the level set function of previous frame according to the displacement information of object. However, failure rates of single paramecium tracking is still 52% after displacement correction. These results are not good enough for biologist application.

Since we have obtained the certain robustness of single paramecium tracking using the deterministic method, therefore, we try to combine the stochastic methods to increase our tracking robustness. Because in our tracking system, disturbances such as measurement errors from camera, measurement error from servo motor of stage and control precision of servo motor, it is hard for us to describe our system as a linear model, we therefore choose particle filter (PF) as our basic framework. The most common PF algorithm, i.e., condensation filter combined with PR-LSM-DC as measurement is proposed to estimate the object location for single moving paramecium tracking. The experiment results indicate that the proposed model successfully improves robustness of tracking performance during collision and prolong the average tracking duration.

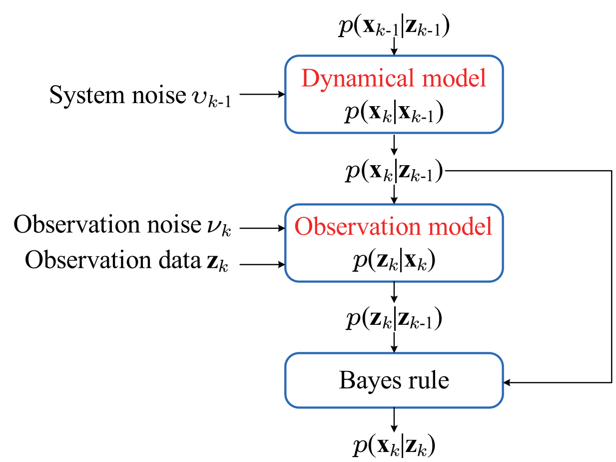


Fig. 1. Updating the posterior density in the iterative computation.

II. OBJECT TRACKING BASED ON PARTICLE FILTER

A. The general object tracking problem

Object tracking based on PF is described as the problem of estimating the state vector $\mathbf{x}_k$ of a system at time k (discrete) while a set of observations $\mathbf{z}_k$ is available over time. The aim is to estimate the posterior density $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ recursively in time, where $\mathbf{z}_{1:k} = \{y_1, y_2, \cdots, y_k\}$. Ultimately it is required to estimate recursively in time some function $f(\mathbf{x}_k)$ of the object state which is the location of the object in our tracking problem.

X. Fei and K. Hashimoto are with Graduate School of Information Sciences, Tohoku University, 6-6-01 Aramaki Aza Aoba, Aoba-ku, Sendai 980-8579, Japan. sandra_fei, koichi@ic.is.tohoku.ac.jp

X. Fei is also with Electrical Engineering College, Guizhou University, Guiyang, Guizhou, 550003, China.

978-1-4244-6676-4/10/$25.00 ©2010 IEEE

As shown in Fig. 1, an iteration step of process starts with a sample set $\mathbf{x}$ representing the a posteriori density $p(\mathbf{x}_{k-1}|\mathbf{z}_{k-1})$ from the previous time step. Then the posterior density $p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})$ is propagated into the next time step via the transition density $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ as follows:

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})d\mathbf{x}_{k-1} \quad (1)$$

The next step involves the application of Bayes rule when new measurement data $p(\mathbf{z}_k|\mathbf{x}_k)$ are observed:

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{z}_{1:k-1})}{p(\mathbf{z}_k|\mathbf{z}_{1:k-1})}. \quad (2)$$

After Bayes rules, sample set now represents the new a posteriori density $p(\mathbf{x}_k|\mathbf{z}_k)$.

The expected value of the function $f(\mathbf{x}_k)$ is computed as

$$E[f_k(\mathbf{x}_k)] = \int f(\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{z}_{1:k})d\mathbf{x}_k. \quad (3)$$

The prediction and update strategy of (1) and (2) provides an optimal solution to the tracking problem, which, unfortunately, involves high-dimensional integration. Therefore, iterative sampling techniques can be used which leads to the use of particle filter.

B. General Particle Filter

The key idea of particle filter is to represent the required posterior probability density function (PDF) by a set of random samples with associated weights and to achieve estimation based on these samples and weights.

To develop the details of particle filter, we define $\{\mathbf{x}^i_{0:k}, \mathbf{w}^i_k\}_{i=1}^N$ as a random measurement set that characterizes the posterior PDF $p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})$, where $\{\mathbf{x}^i_{0:k}, i = 0, \cdots, N\}$ is a set of support points with associate weights $\{\mathbf{w}^i_k, i = 0, \cdots, N\}$. $\mathbf{x}_{0:k} = \{x_j, j = 0, \cdots, k\}$ is the set of all states up to time k. The weights are normalized as $\sum_i \mathbf{w}^i_k = 1$. The posterior density at k can be approximated as

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) \approx \sum_{i=1}^{N} \mathbf{w}^i_k \delta(\mathbf{x}_{0:k} - \mathbf{x}^i_{0:k}) \quad (4)$$

The weights are chosen using the principle of importance sampling [5], [6]. Let $\mathbf{x}^i \sim q(x), i = 1, \cdots, N$ be samples that are easily generated from a proposal distribution $q(\cdot)$ called an importance density. Then, the weight update equation can be shown to be

$$\mathbf{w}^i_k = \mathbf{w}^i_{k-1} \frac{p(\mathbf{z}_k|\mathbf{x}^i_k)p(\mathbf{x}^i_k|\mathbf{x}^i_{k-1})}{q(\mathbf{x}^i_k|\mathbf{x}^i_{0:k-1}, \mathbf{z}_{1:k})}. \quad (5)$$

The posterior density is non-parametrically approximated according to weighted states of particles. Then, the mean state of the particles is treated as the estimated value which we are interested in.

C. Condensation Filter

The Condensation algorithm, a variant of particle filter, evolves system states according to their probabilities which are calculated from the observations [7]. Recently, the Condensation algorithm has been introduced for non-Gaussian, nonlinear contour tracking problems [8], [9]. The Condensation generates a weighted, time-stamped sample set, denoted by $\{\mathbf{x}^i_k, \mathbf{w}^i_k, i = 1, \cdots, N\}$, to approximate the conditional state-density $p(\mathbf{x}_k|\mathbf{z}_k)$ at time k. Its iterative process is depicted as following:

1) Initializing particles from the prior $p(\mathbf{x}^i_0)$ to obtain a set $\{\tilde{\mathbf{x}}^i_0, 1/N, i = 1, \cdots, N\}$.
2) Predicting particles $p(\mathbf{x}_k|\mathbf{x}_{k-1} = \tilde{\mathbf{x}}_k)$ using dynamic model.
3) Weighting
   a. Weight the new state in terms of the measured features $\mathbf{z}_k$:
   $$\mathbf{w}^i_k = p(\mathbf{z}_k|\mathbf{x}_k = \mathbf{x}^i_k). \quad (6)$$
   b. Normalize the particle weights so that $\sum_n \mathbf{w}^i_k = 1$.
   $$\tilde{\mathbf{w}}^i_k = \frac{\mathbf{w}^i_k}{\sum_{i=1}^{N} \mathbf{w}^i_k} \quad (7)$$
   c. Store normalized particle weights together as cumulative probability $\mathbf{c}^i_k$ where
   $$\begin{aligned} \mathbf{c}^0_k &= 0, \\ \mathbf{c}^i_k &= \mathbf{c}^{i-1}_k + \tilde{\mathbf{w}}^i_k, i = 1, \cdots, N, \\ \mathbf{c}'^i_k &= \frac{\mathbf{c}^i_k}{\mathbf{c}^i_N}, i = 1, \cdots, N. \end{aligned} \quad (8)$$
4) Outputting a set of particles $\{(\mathbf{x}^i_k, \mathbf{w}^i_k), i = 1, \cdots, N\}$ that can be used to approximate the posterior distribution:
   $$p(\mathbf{x}_k|\mathbf{z}_{1:k}) \approx \sum_{i=1}^{N} \mathbf{w}^i_k \delta(\mathbf{x}_k - \mathbf{x}^i_k). \quad (9)$$
5) Resampling
   a. Generate a sequence of N sorted random number T_r uniformly distributed in $[0, 1]$.
   b. Find the smallest j for which $c'^j \geq T^j_r$
   c. Set $\mathbf{x}^i_k = \mathbf{x}^j_k$.
6) $k = k + 1$, go to step 2.

III. SINGLE PARAMECIUM TRACKING BASED ON CONDENSATION

A. Coordinate System

For single paramecium tracking, we are interested in localizing target to keep target in the center of visual field under microscope by moving stage in opposite direction. It is important to analyze the movement of paramecium in the tracking system. As we can see in Fig. 2, the stage is moved within control table by controlling motor. A coordinate system for stage moving on control table is defined as in

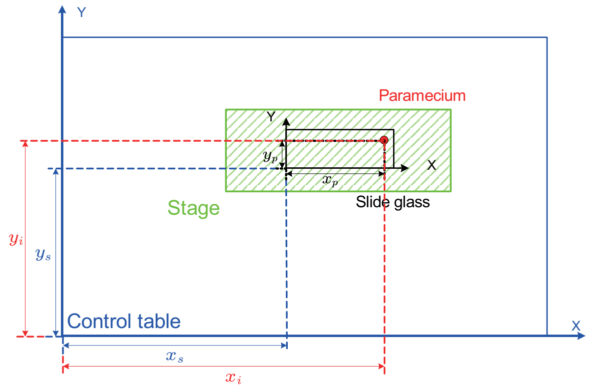


Fig. 2. Coordinate system for single paramecium tracking.

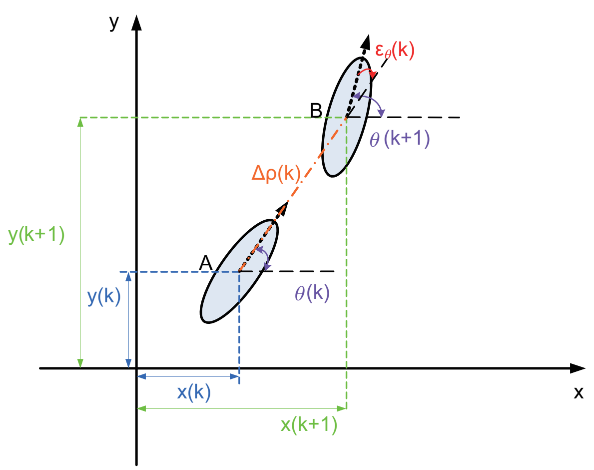


Fig. 3. Schematic diagram of paramecium motion with translation and rotation information.

blue and the location of stage in stage coordinate system is represented as (x_s, y_s). The slide glass shown as black rectangle in Fig. 2 is fixed on the stage and moving with the stage together. Paramecium shown as red dot in Fig. 2 swims within slide glass. A coordinate system for paramecium moving on slide glass is defined as in black and the location of paramecium in slide glass coordinate system is represented as (x_p, y_p). The motion of stage relative to control table combining with the motion of paramecium relative to slide glass results in the motion of paramecium in the image that is captured by the camera mounted on microscope. A coordinate system for paramecium moving on the image is defined as in red and the location of paramecium in image coordinate system is represented as (x_i, y_i).

B. Motion of Paramecium

In order to predict the probability distribution of the pose of the moving paramecium we propose a motion model which describes translation and rotation information of the moving target cell. The motion of a cell from time k to $k+1$ is shown in Fig. 3. The simplest approximation of the moving process is to model this motion as a translation along its own axis followed by a rotation. The orientation of the cell at the beginning (location A in Fig. 3) denoted by $\theta(k)$ and the orientation at the end (location B in Fig. 3) by $\theta(k+1) = \theta(k) + \varepsilon_\theta$, where ε_θ represents the amount of the rotation that occurs after the translation modeling by the effect of noise [10], [11], [12]. The translation $\Delta\rho(k)$ from time k to $k+1$ is modeled by the translation from time k to $k-1$ plus unpredictable noise. Then the paramecium motion model can be represented as eq. 10.

$$\begin{aligned} x(k+1) &= x(k) + \Delta\rho(k)\cos(\theta(k)) + \varepsilon_x(k) \\ y(k+1) &= y(k) + \Delta\rho(k)\sin(\theta(k)) + \varepsilon_y(k) \\ \theta(k+1) &= \theta(k) + \varepsilon_\theta \\ \Delta\rho(k) &= \sqrt{(x(k)-x(k-1))^2 + (y(k)-y(k-1))^2}, \end{aligned} \tag{10}$$

ε_θ indicates rotation angle and ε_x and ε_y indicates unpredictable velocity change.

C. System Dynamical Model

As we can see in Fig. 4, input for tracking system is the differences between reference location (x_r, y_r) and paramecium location on image (x_i, y_i). (x_r, y_r) is the desired location where the target needs to be kept. The paramecium location on image (x_i, y_i) can be measured from our boundary detection algorithm which is the consequences of paramecium moving on the slide glass and stage moving on the control table. The state vector for tracking system at time step k is defined as containing the location of paramecium on slide glass at time step $k+1$ and location of stage on control table at time step $k+1$ and it is therefore written as

$$\mathbf{x}_k^T = (x_p(k+1), y_p(k+1), \theta(k+1), x_s(k+1), y_s(k+1))^T. \tag{11}$$

The output vector is defined as

$$\mathbf{y}_k^T = (x_i(k), y_i(k))^T. \tag{12}$$

System models are built as non-linear state dynamics with non-linear disturbances:

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{f}(\mathbf{x}_k) + \mathbf{B_u}\mathbf{u}_k + \upsilon_k \\ \mathbf{y}_k &= \mathbf{g}(\mathbf{x}_k) + \nu_k \\ \mathbf{u}_k &= \mathbf{k}(\mathbf{r} - \mathbf{y}_k) \end{aligned} \tag{13}$$

Here $\mathbf{x}_k$ is the state vector. $\mathbf{f}$ and $\mathbf{g}$ are functions for states and measurements. $\mathbf{u}_k$ is measured inputs. υ_k represents unpredictable system disturbances. $\mathbf{y}_k$ is the measurements and ν_k is the measurements noises. The resulting system model is

$$\begin{aligned} \begin{pmatrix} x_p(k+1) \\ y_p(k+1) \\ \theta(k+1) \\ x_s(k+1) \\ y_s(k+1) \end{pmatrix} &= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_p(k) \\ y_p(k) \\ \theta(k) \\ x_s(k) \\ y_s(k) \end{pmatrix} \\ &+ \begin{pmatrix} \Delta\rho(k)\cos(\theta(k)) \\ \Delta\rho(k)\sin(\theta(k)) \\ 0 \\ 0 \\ 0 \end{pmatrix} + \mathbf{B_u}\mathbf{u} + \begin{pmatrix} \varepsilon_x(k) \\ \varepsilon_y(k) \\ \varepsilon_\theta(k) \\ 0 \\ 0 \end{pmatrix} \end{aligned} \tag{14}$$

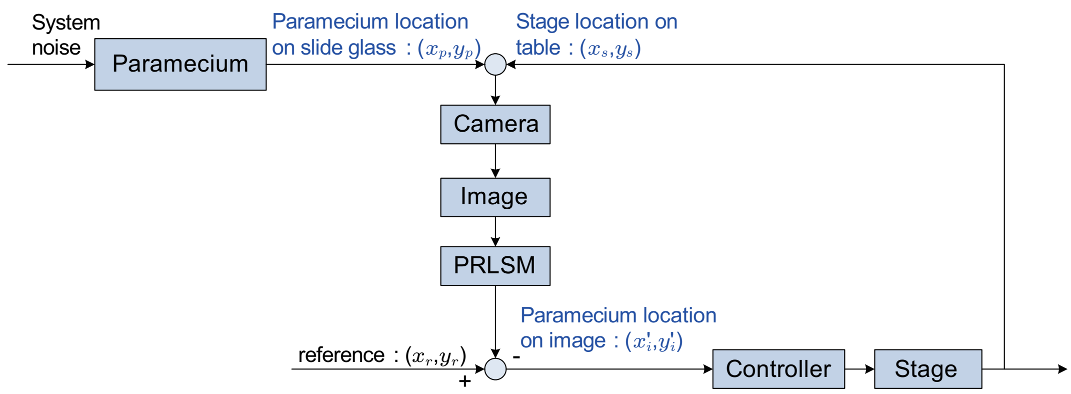


Fig. 4. Schematic diagram of dynamic for single paramecium tracking.

$$\begin{pmatrix} x_i(k) \\ y_i(k) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} x_p(k) \\ y_p(k) \\ \theta(k) \\ x_s(k) \\ y_s(k) \end{pmatrix} + \begin{pmatrix} \nu_x(k) \\ \nu_y(k) \end{pmatrix} \tag{15}$$

$$\mathbf{B_u} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{16}$$

$$\mathbf{u} = \begin{pmatrix} k_x & 0 \\ 0 & k_y \end{pmatrix} \left[\begin{pmatrix} x_r \\ y_r \end{pmatrix} - \begin{pmatrix} x_i(k) \\ y_i(k) \end{pmatrix} \right] \tag{17}$$

where k_x and k_y are gain parameters for proportional control.

D. Observation Model

We uses detected contour from PR-LSM-DC as the measurement cue. Based on detected contour, we can calculate the centroid of current detected object. Location observation model of target on image is proposed as

$$\begin{aligned} p(\mathbf{y}_k|\hat{\mathbf{x}}_k) \propto & \frac{1}{\sqrt{2\pi}\sigma_c} \exp\left(-\frac{(x_i(k)-\hat{x}_i(k))^2 + (y_i(k)-\hat{y}_i(k))^2}{2\sigma_c^2}\right) \\ & \frac{1}{\sqrt{2\pi}\sigma_\theta} \exp\left(-\frac{(\theta_i(k)-\hat{\theta}_i(k))^2}{2\sigma_\theta^2}\right), \end{aligned} \tag{18}$$

where $(x_i(k), y_i(k)), \theta_i(k)$ is the pose of paramecium measured by using PR-LSM-DC, $(\hat{x}_i(k), \hat{y}_i(k), \hat{\theta}_i(k))$ is the pose of paramecium estimated using system dynamical model. $\sigma_c = 1$ is the variance keeping all the particles within a circle (center is $(x_i'(k), y_i'(k))$, radius is 1) and σ_θ the variance keeping all the the rotation angle for each particles not much different from measured one.

E. Tracking Based on Condensation

Our proposed method works iteratively as follows:

1) Prepare the image $I(x,y)$ of the tracked object (image size is $s \times s$.) and initialize the particle-set $\{\mathbf{x}_0^i, \mathbf{w}_0^i\}, i = 1, \cdots, N$ where

$$\mathbf{w}_0^i = \frac{1}{N} \tag{19}$$

$$\mathbf{x}_0^i = \begin{pmatrix} x_p(0) \\ y_p(0) \\ \theta(0) \\ x_s(0) \\ y_s(0) \end{pmatrix} = \begin{pmatrix} \frac{s}{N} i \\ \frac{s}{N} i \\ \frac{360}{N} \\ x_p \\ x_p \end{pmatrix} \tag{20}$$

2) By using system dynamic models above, we predict the location of target paramecium on image coordinate system $(\hat{x}_i(k), \hat{y}_i(k)), \hat{\theta}_i(k)$ at the next time-step which helps correct boundary detection error due to the collision.

$$\begin{aligned} \hat{\mathbf{x}}_{k+1} &= \mathbf{f}(\hat{\mathbf{x}}_k) + \mathbf{B_u}\mathbf{u}_k + \hat{\upsilon}_k \\ \hat{\mathbf{y}}_k &= \mathbf{g}(\hat{\mathbf{x}}_k) + \hat{\nu}_k \end{aligned} \tag{21}$$

We assume $\hat{x}_0$ known and $\Delta\rho(k)$ inside function $\mathbf{f}$ is fixed as certain value for presenting paramecium velocity. In $\hat{\upsilon}_k$, we assume noise ε_x and ε_y to be uniformly distributed randoms indicating unpredictable velocity change within [-0.1, 0.1] micrometer. ε_θ is assumed to be randoms drawn from uniform distribution indicating rotation angle within [-0.5, 0.5] degree. ν_k is the measurements noise which is drawn from uniformly distributed randoms within [-1, 1] pixel.

3) Using PR-LSM-DC model, detected contour is obtained. When collision happens, we do not just translate previous $\phi(k-m)$ according to displacement, we also rotate previous ϕ according to $\theta(k) - \theta(k-m)$. Based on the ϕ of corrected contour, the centroid $(x_i(k), y_i(k))$ of detected object is calculated as:

$$\begin{aligned} m_{pq} &= \sum_x \sum_y x^p y^q I_\phi(x,y). \\ x_i(k) &= \frac{m_{10}}{m_{00}} \\ y_i(k) &= \frac{m_{01}}{m_{00}}. \\ \theta(k) &= \frac{1}{2} tan^{-1} \frac{2\mu_{11}}{\mu_{20} - \mu_{02}}, \end{aligned} \tag{22}$$

where μ_{11}, μ_{20} and μ_{02} are first-oder and second-order central image moments at time step k. Then particle weights are updated using observation model (18) as:

$$\mathbf{w}_k^i = p(\mathbf{y}_k|\hat{\mathbf{x}}_k).$$

where $\sigma_c = 1$, $\sigma_\theta = 0.5$.

4) Once the N particle set have been estimated, the location of single target paramecium at time-step k is calculated as:

$$E[f(\mathbf{x}_k)] = \sum_{i=1}^{N} \mathbf{w}_k^i \hat{\mathbf{x}}_k^i. \tag{23}$$

5) Calculate the normalized particle weights $\tilde{\mathbf{w}}_k^i$ and cumulative probability $\mathbf{c}_k^i$. Check out if resampling is necessary.
6) $k = k+1$, go to step 2.

IV. EXPERIMENTS

A. Successfully tracking during Collision

To confirm the ability of our proposed model, we conduct experiments of tracking the single paramecium with condensation in a non-parallel workstation. We check whether our proposed method track only target paramecium even when the tracked paramecium collides with other obstacles. Fig. 5(a) and Fig. 5(b) are consecutive image sequences from the tracking movies. The center of target paramecium is calculated by using condensation filter with PR-LSM-DC as measurements shown as red dot in Fig. IV-A.

Fig. 5(a) is the result of tracking the single paramecium collided with the other paramecium. At 62 [ms], the target paramecium without collision is in the center of the image. At 71 [ms], the target paramecium collides with the other paramecium. However, the detected location of target does not slide to another paramecium and the target paramecium is kept around the center of the image.

Fig. 5(b) is the result of tracking the single paramecium near a air bubble. From 371 [ms] to 377 [ms], the target paramecium swims close to the air bubble. During 379 [ms] to 387 [ms], the tracked paramecium collides with the air bubble. After 387 [ms], this paramecium swims away from the air bubble. During the whole process of collision, the target paramecium is in the center of the image all the time, indicating that the single paramecium tracking is successfully using condensation filter.

B. Success and Failure Reasons

To clarify how our proposed collision handling improves the tracking robustness, we compare success and failure rates of tracking with condensation filter to the ones without condensation filter. 50 trails of real-time tracking are conducted for two situations. We define successful tracking and failure tracking as the duration of paramecium staying in the image is over 60 s or not, respectively. The success rate of tracking increases from 48% to 66% due to the condensation.

The reasons of failure tracking are (1) collide with lots of obstacles, (2) lose focus and (3) limitation of system. In case of reason (1), target moves into a high density cell population, the boundary of the target contacts or overlaps with others, our boundary detection will fail in distinguishing which area is the target. For the reason (2), tracking in 2D is guaranteed by limiting the height of water pool for paramecium swimming. If our tracking target happens to be a small one, the body of this cell might loses focus partially due to the body sinking. This problem interests us as a 3D cell tracking. The reason (3) presents as limitation of stage moving range which can be adjusted according to specific biologic experiment requirements.

C. Tracking Duration

We compare maximum, minimum and average durations of the real time tracking trials with and without condensation to show how the collision handling increase the tracking duration (Tab. I). The maximum tracking duration is increased up to 846 [ms] due to our proposed filter. The minimum tracking durations with and without condensation are 4 [ms] and 2 [ms] respectively, indicating there is no big difference between them. If target paramecium is among high density population, usually tracking will fail in very short time. This is a problem that is still not solved even with condensation. The average tracking duration is increased up to 191 [ms] by using the condensation filter. These increases of maximum and average tracking durations are valuable to observe a moving cell under the microscope for long time.

TABLE I

MAXIMUM, MINIMUM AND AVERAGE DURATIONS OF TRACKING TRAILS WITH AND WITHOUT CONDENSATION FILTER.

Duration	With condensation	Without condensation
Maximum [s]	846	523
Minimum [s]	4	2
Average [s]	191	74

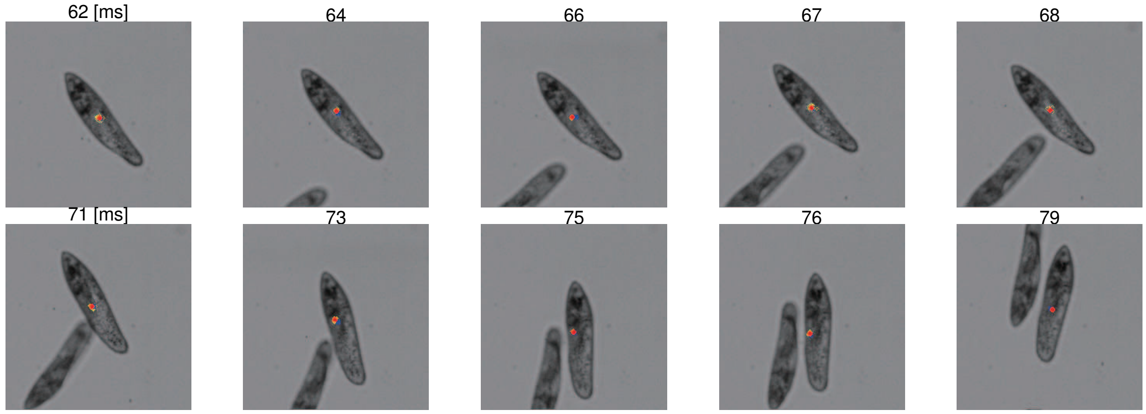
D. Computational Time

To verify whether condensation filter implemented in workstation is capable of real-time tracking, we compute an average computational time for each frame from 10 image sequences (Tab. II). The average time for 1st frame is 3.15 [ms]. The average time for other frames is 2.13 [ms]. This reduction of is because level set function $\phi^{m,0}$ used in PR-LSM-DC is initialized as the converged ϕ of the last frame and the position change of the tracked paramecium between two consecutive frames is very small. These results represent that our method can provide about 2 [ms] cycle time of the single cell tracking with collision handling for the real-time tracking.

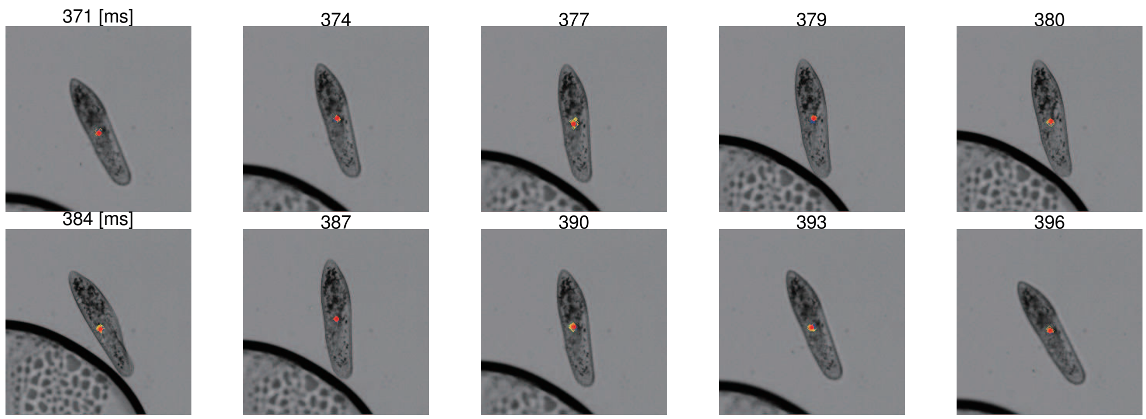
TABLE II

AVERAGE COMPUTATIONAL TIME FOR THE CONVERGENCE IN REAL-TIME SINGLE PARAMECIUM TRACKING.

Content	Workstation
one loop time: 1st	3.15 [ms]
one loop time: others	2.13 [ms]



(a) Location of target (colliding with others)



(b) Location of target (colliding with bubbles)

Fig. 5. Locations of target paramecium detected using condensation with PR-LSM-DC in nonparallel PC.

V. CONCLUSION AND DISCUSSION

To improve the robustness of single paramecium tracking further, we combined condensation filter with PR-LSM-DC model. To implementing this combined model, a paramecium motion model, a tracking system dynamic model and an observation model was established. Considering the small differences between two frames, we built up a simple linear model for paramecium motion under microscope. The evolving system states were estimated from dynamical model. Using our previous PR-LSM-DC model as measurements, the observation model updated weights for all the states in PF. Experiments confirmed that with the motion prediction from condensation, this proposed method also increases maximum tracking durations, average tracking durations, and the success rate of single cell tracking among other obstacles. However, still cannot solve the tracking problem when target among large cell population.

REFERENCES

[1] X. Fei, Y. Igarashi, and K. Hashimoto. 4ms level set method - parallel implementation of contour detection of paramecia by using cpv system. *17th CISM-IFToMM Symposium on Robot Design, Dynamics, and Control*, pages 27–34, 2008.

[2] K. Hashimoto X. Fei, Y. Igarashi. A parallel computation of the region-based level set method for boundary detection with moving objects. *Journal of Robotics and Mechatronics*, 21(6):698–708, December 2009.

[3] X. Fei, Y. Igarashi, and K. Hashimoto. 2d tracking of single paramecium by using parallel level set method and visual servoing. *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 752–757, July 2008.

[4] Koichi Hashimoto Xianfeng Fei, Yasunobu Igarashi. Parallel region-based level set method with displacement correction for tracking a single moving object. In *Advanced Concepts for Intelligent Vision Systems, Bordeaux, France*, pages 27–34, 2009.

[5] A. Doucet, S. J. Godsill, and C. Andrieu. *On sequential simulation-based methods for Bayesian filtering*. Citeseer, 1998.

[6] A. Doucet, S. Godsill, and C. Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and computing*, 10(3):197–208, 2000.

[7] U. Grenander, Y. Chow, and D.M. Keenan. *Hands: A Pattern Theoretic Study of Biological Shapes. Research notes in Neural Computing*. Springer Verlag, New York, 1991.

[8] M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. *Lecture Notes in Computer Science*, 1064:343–356, 1996.

[9] M. Isard and A. Blake. Condensation-conditional density propagation for visual tracking. *International journal of computer vision*, 29(1):5–28, 1998.

[10] I. M. Rekleitis. A particle filter tutorial for mobile robot localization. In *International Conference on Robotics and Automation*, volume 3, pages 1–36, 2003.

[11] J. Borenstein, HR Everett, L. Feng, et al. *Navigating mobile robots: Systems and techniques*. AK Peters, 1996.

[12] J. Borenstein and L. Feng. Measurement and correction of systematic odometry errors in mobile robots. *IEEE Transactions on Robotics and Automation*, 12(6):869–880, 1996.