Presentation date: March, 12, 2010 Paper number: 256-4

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

○ Xianfeng Fei Koichi Hashimoto

Graduate School of Information Sciences, Tohoku University, sandra_fei, koichi@ic.is.tohoku.ac.jp
6-6-01 Aramaki Aza Aoba, Aoba-ku, Sendai 980-8579, Japan, Tel: 22-795-7018

In this paper, to increase success rate of single paramecium tracking further, we combine condensation filter with region-based level set method model. For the condensation filter, a system dynamical model is built up to estimate the location of target paramecium on image coordinate system. And also an observation model based on location and intensity summation of target is created to update particle weights. Experiments confirmed that with the motion prediction from condensation, we extend the single paramecium tracking duration and increase the success rate of tracking.

Key Words: particle filter, condensation, level set method, visual servoing.

1 INTRODUCTION

Failure rates of single paramecium tracking is still 52% after displacement correction [1]. These results are not good enough for biologist application. So far, we have increased robustness of single paramecium tracking by decrease boundary detection error after error happened. Therefore, we try to use some prediction technique to increase our Success rates. Therefore, we try to combine particle filter with our previous work to increase robustness of single paramecium tracking during collision.

Since 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 as our basic framework. Condensation filter is used in this chapter to calculate the target centroid based on estimated location using dynamic model and measurements from PR-LSM. The proposed model is implemented in workstation (a nonparallel high speed computer). The experiment results indicate that condensation filter is helpful for the collision problem and raise success rate of tracking and prolong the maximum tracking duration.

2 TRACKING BASED ON PF

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. 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. Particle filter represents the required posterior probability density function (PDF) as a set of random samples with associated weights and to compute estimates based on these samples and weights. The Condensation algorithm, one of the most common particle filter, is based on factored sampling which has been introduced for non-Gaussian, nonlinear contour tracking problems [2, 3, 4]. Given that the process at each time-step is a self-contained iteration of factored sampling, the output of an iteration will be a weighted, time-stamped sample set, denoted $\{\mathbf{x}_k^i, i = 1, \cdots, N\}$ with weights $\mathbf{w}_k^i$, representing approximately the conditional state-density $p(\mathbf{x}_k|\mathbf{z}_k)$ at time k.

3 TRACKING BASED ON CONDENSATION

3.1 Motion of Paramecium

In order to predict the probability distribution of the pose of the moving paramecium after a motion we propose a motion model which describes translation and rotation information of target cell. The motion of a cell from time k to $k+1$ is shown in Fig. 3.1. 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. 3.1) would be $\theta(k)$ and the orientation at the end (location B in Fig. 3.1) is $\theta(k+1) = \theta(k) + \varepsilon_\theta$, where ε_θ represents the amount of the rotations that occur after the translation modeling by the effect of noise [5, 6, 7]. 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. 1.

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

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

3.2 System Dynamical Model

In single paramecium tracking system we are interested in localizing target to keep target in the center of visual field under microscope by moving stage in opposite direction. The stage moves within control table by controlling motor. A coordinate system for stage moving on control table is defined and the location of stage in stage coordinate system is represented as (x_s, y_s). The slide glass is fixed on the stage and moving with the stage together. Paramecium is swimming within slide glass. A coordinate system for paramecium moving on slide glass is defined and the location of paramecium in slide glass coordinate system is represented as $(x_p.y_p)$. The motion of stage on control table combining with the motion of paramecium on 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 and the location of paramecium in image coordinate system is represented as (x_i, y_i).

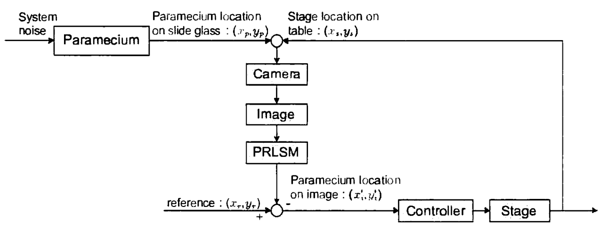

Figure 1: Schematic diagram of dynamic for single paramecium tracking.

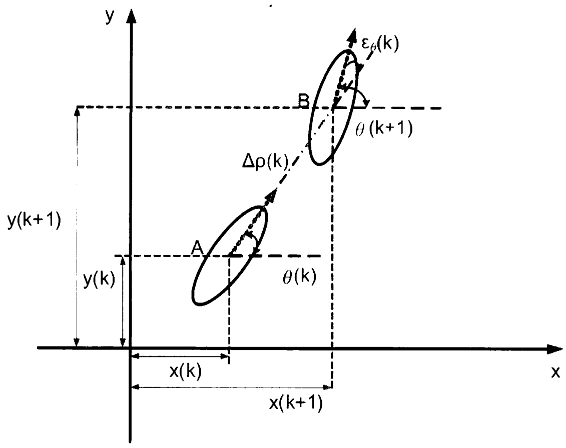

Figure 2: Schematic diagram of paramecium motion with translation and rotation information.

As we can see in Fig. 1, 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 is 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

$$\begin{aligned} \mathbf{x}_k^T =& (x_p(k+1), y_p(k+1), \theta(k+1), x_s(k+1), \\ & y_s(k+1))^T. \end{aligned} \tag{2}$$

The output vector is defined as

$$\mathbf{y}_k^T = (x_i(k).y_i(k))^T. \tag{3}$$

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

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{f}(\mathbf{x}_k) + \mathbf{B}_\mathbf{u}\mathbf{u}_k + \upsilon_k \\ \mathbf{y}_k &= \mathbf{g}(\mathbf{x}_k) + \nu_k \\ \mathbf{u}_k &= \mathbf{k}(\mathbf{r} - \mathbf{y}_k) \end{aligned} \tag{4}$$

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 is the noise model representing unpredictable change of paramecium translation. $\mathbf{y}_k$ is the measurements and ν_k is the measurements noises. The resulting system model is

$$\begin{pmatrix} x_p(k+1) \\ y_p(k+1) \\ \theta(k+1) \\ x_s(k+1) \\ y_s(k+1) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_p(k) \\ y_p(k) \\ \theta(k) \\ x_s(k) \\ y_s(k) \end{pmatrix} + \begin{pmatrix} \Delta\rho(k)\cos(\theta(k)) \\ \Delta\rho(k)\sin(\theta(k)) \\ 0 \\ 0 \\ 0 \end{pmatrix} + \mathbf{B_u u} + \begin{pmatrix} \varepsilon_x(k) \\ \varepsilon_y(k) \\ \varepsilon_\theta(k) \\ 0 \\ 0 \end{pmatrix} \tag{5}$$

$$\begin{pmatrix} x_i(k) \\ y_i(k) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} x_p(k) \\ y_p(k) \\ \theta(k) \\ x_s(k) \\ y_s(k) \end{pmatrix} + \begin{pmatrix} v_x(k) \\ v_y(k) \end{pmatrix} \tag{6}$$

$$\mathbf{B_u} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{7}$$

$$\mathbf{u} = \begin{pmatrix} k_x & 0 \\ 0 & k_y \end{pmatrix} \left[\begin{pmatrix} x_r \\ y_r \end{pmatrix} - \begin{pmatrix} x_i(k) \\ y_i(k) \end{pmatrix} \right] \tag{8}$$

where k_x and k_y are gain parameters for proportional control.

3.3 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 objec. Location observation model of target on image is proposed as

$$\begin{aligned} p(\mathbf{y}_k|\hat{\mathbf{x}}_k) \propto & \frac{1}{\sqrt{2\pi}\sigma_c}\exp(-\frac{(x_i(k)-\hat{x}_i(k))^2)^2}{2\sigma_c^2} \\ & + \frac{(y_i(k)-\hat{y}_i(k))^2}{2\sigma_c^2}) \\ & \frac{1}{\sqrt{2\pi}\sigma_\theta}\exp(-\frac{(\theta_i(k)-\hat{\theta}_i(k))^2}{2\sigma_\theta^2}), \end{aligned} \tag{9}$$

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. Then particle weights can be updated based on this likelihood as:

$$\mathbf{w}_k^i = p(\mathbf{y}_k|\hat{\mathbf{x}}_k). \tag{10}$$

3.4 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

$$\mathbf{w}_0^i = \frac{1}{N} \tag{11}$$

$$\mathbf{x}_0^i = \begin{pmatrix} x_p(0) \\ y_p(0) \\ \theta(0) \\ x_s(0) \\ y_s(0) \end{pmatrix} = \begin{pmatrix} \frac{s}{N}i \\ \frac{s}{N}i \\ \frac{360}{N} \\ x_p \\ x_p \end{pmatrix} \tag{12}$$

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.

$$\begin{aligned} \hat{\mathbf{x}}_{k+1} &= \mathbf{f}(\hat{\mathbf{x}}_k) + \mathbf{B_u u}_k + \hat{\upsilon}_k \\ \hat{\mathbf{y}}_k &= \mathbf{g}(\hat{\mathbf{x}}_k) + \hat{v}_k \end{aligned} \tag{13}$$

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. v_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 particle weights are updated using observation model (9) 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:

$$E[f(\mathbf{x}_k)] = \sum_{i=1}^{N} \mathbf{w}_k^i \hat{\mathbf{x}}_k^i. \tag{14}$$

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.

4 EXPERIMENTS

4.1 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. 4.1 is 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. 4.1.

Fig. 4.1 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.

4.2 Success and Failure Reasons

To clarify how our proposed collision handling improves the tracking success rate, we compare success and failure rates of tracking with condensation filter to the ones without condensation filter (Table 1). 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 listed as in Tab. 1. For reason (1), the failure rate is improved significantly by using condensation, suggesting that condensation improves the robustness of the single paramecium tracking among a few obstacles. For reason (2), the failure rate with and without condensation are nearly equal, that indicating condensation filter can not solve the problem with large population of cells.

5 CONCLUSION AND DISCUSSION

To increase success rate of single paramecium tracking further, we combine condensation filter with PR-LSM-DC model. For the condensation filter, a system dynamical model is built up to estimate the location of target paramecium on image coordinate system. And also an observation model based on location and intensity summation of target calculated from PR-LSM-DC model is created to update particle weights. 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] 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*, 27–34, 2009.

[2] 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.

[3] M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional den-

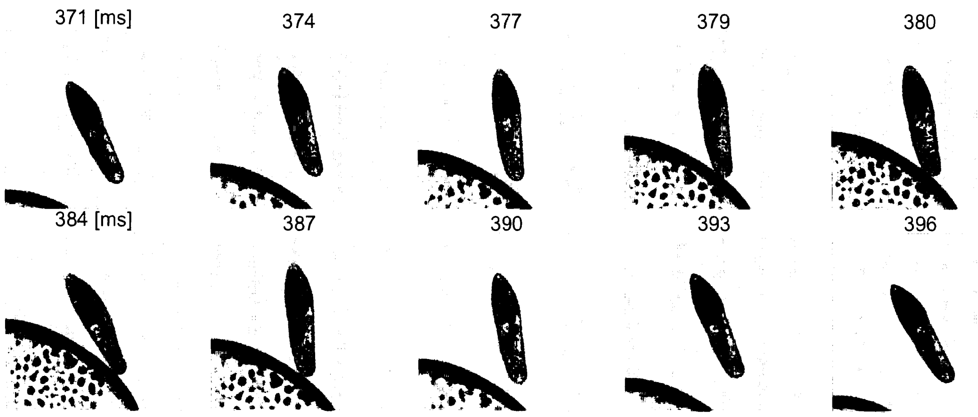

Figure 3: Locations of target paramecium detected using condensation with PR-LSM-DC in nonparallel PC.

Table 1: Success and failure rates of paramecium tracking trails with and without condensation filter. Reasons for tracking failure and their frequencies are listed.

Result	With condensation	Without condensation
Success	33/50(66%)	24/50(48%)
failure	17/50(34%)	26/50(52%)
(1) Collision (obstacles ≤ 3)	10/50(20%)	15/50(30%)
(2) Collision (obstacles > 3)	6/50(12%)	8/50(16%)
(3) Losing focus	1/50(2%)	2/50(4%)
(4) Limitation of the system	0/50(0%)	1/50(2%)
(5) Unknown	1/50(2%)	0/50(0%)

sity. *Lecture Notes in Computer Science*, 1064:343–356, 1996.

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

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

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

[7] 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.