Voting-based visual tracking with temporal peak propagation

Kazuhiko Kawamoto and Kaoru Hirota
Interdisciplinary Graduate School of Science and Engineering, Tokyo Institute of Technology
Mail-Box:G3-49, 4259 Nagatsuta, Midori-ku, Yokohama 226-8502, Japan
Email: kawa@hrt.dis.titech.ac.jp

***Abstract*— A visual tracking algorithm in a sequence of cluttered binary images is proposed using a voting procedure and a sequential Monte Carlo (SMC) filter. The proposed algorithm robustly detects curves in binary images using the voting, and propagates the results obtained by the voting in the image at time t to the successive image at time $t+1$ using the SMC filter. This temporal propagation enables to focus on the potential region, where the curves detected at time $t-1$ are likely to appear at time t, thus being able to efficiently search the curves at time t. Furthermore, a particle elimination procedure is introduced to prevent the variance of the filtering distribution from increasing over time. To evaluate the tracking performance in clutter, two noisy image sequences, consisting of 90 frames witch 320×240 pixels, are used. The results shows that the algorithm successfully tracks the circle and the circular arc in the noisy binary images and that the execution time per frame for each image sequence is 23.6 msec and 23.5 msec, respectively, on the average of 10 trials. The two combinations of the voting and the SMC filtering give a basis of robust and efficient image processing, which has recently attracted increasing attention from academic and industry.**

***Keywords*: Voting, Visual Tracking, Sequential Monte Carlo Filter, State Space Model**

I. Introduction

Object tracking in a sequence of images has various applications, such as preprocessing of motion analysis, object recognition, and video compression, in the fields of intelligent image processing and computer vision. In particular, robust and efficient methods for tracking objects in clutter attracts increasing attention from academic and industry.

For a few decades, voting-based algorithms, such as the Hough transform [1], RANSAC(RANdom SAmple Consensus) [2], and their variants[3][4][5][6], have been widely used to detect objects, because their algorithms are robust even in cluttered images. However, the voting-based algorithms are, in general, time-consuming, thus being unsuitable for real-time applications.

This paper propose a robust and efficient voting-based algorithm for tracking curves in a sequence of cluttered images. The proposed algorithm propagates votes, which indicate the candidates of curves, obtained in an image at time t to the successive image at time $t+1$ using a sequential Monte Carlo (SMC) filter [7][8][9][10]. Since the SMC filters approximate a continuous state distribution by discrete finite sample points, called particles, the voting-based algorithm with temporal peak propagation is naturally implemented by assigning each sample point of the distribution to the parameters of a curve in the image. Due to temporal peak propagation, the algorithm can focus on the potential region, where the curves detected at the previous time step are likely to appear, thereby searching the curves in the region efficiently. Furthermore, a particle elimination procedure is introduced to suppress rapid diffusion of particles over time, which phenomenon is likely to happen in clutter. To evaluate the tracking performance of the proposed algorithm in clutter, two noisy image sequences, consisting of 90 frames with 320×240 pixels, are used. The results shows that the proposed algorithm successfully tracks the circle and the circular arc in the binary images and that the execution time per frame for each image sequence is 23.6 msec and 23.5 msec, respectively, on the average of 10 trials.

This paper is organized as follows. Sec. II reviews a voting procedure for detecting curves in images. Sec. III presents a basic algorithm of the SMC filters, called the bootstrap filter [7]. Sec. IV proposes the voting-based algorithm for visual tracking with temporal peak propagation. Sec. V reports the experimental results for the two noisy image sequences.

II. Voting-Based Curve Detection in Binary Image

Consider the problem of detecting curves, denoted by $f(\boldsymbol{r};\boldsymbol{a}) = 0$, in a given binary image, where $\boldsymbol{r}$ is a two-dimensional point in the binary image and $\boldsymbol{a}$ is the parameter vector representing the curve. To solve the problem, voting-based algorithms, such as the Hough transform [1] and RANSAC (RANdom SAmple Consensus) [2], first count the number of the points, $I(\cdot)$, on the curve

$$I(\boldsymbol{a}) = \sum_i h(\boldsymbol{r}_i, \boldsymbol{a}), \quad (1)$$

$$h(\boldsymbol{r}_i, \boldsymbol{a}) = \begin{cases} 1 & \text{if} \quad f(\boldsymbol{r}_i; \boldsymbol{a}) = 0, \\ 0 & \text{otherwise}, \end{cases} \quad (2)$$

and then detect the parameter $\boldsymbol{a}$ as the curve existing on the image if $I(\boldsymbol{a}) > T$ for a given threshold $T (\in \boldsymbol{N})$.

There are two typical ways of implementing the counting procedure: the Hough transform [1] and RANSAC [2]. The Hough transform accumulates the number of the curves, $f(\boldsymbol{a};\boldsymbol{r}) = 0$, passing through $\boldsymbol{a}$ in the parameter space for every point $\boldsymbol{r}$ in the image. On the other hand, RANSAC counts the number of the points on a curve by scanning the points along the curve, $f(\boldsymbol{r};\boldsymbol{a}) = 0$, for every parameter $\boldsymbol{a}$ in the image. In general, RANSAC requires less computer storage than the Hough transform; the amount of storage for the Hough transform exponentially increases with the dimension of the parameter. Thus, RANSAC is more suitable for curves described by high dimensional parameters. Indeed, RANSAC

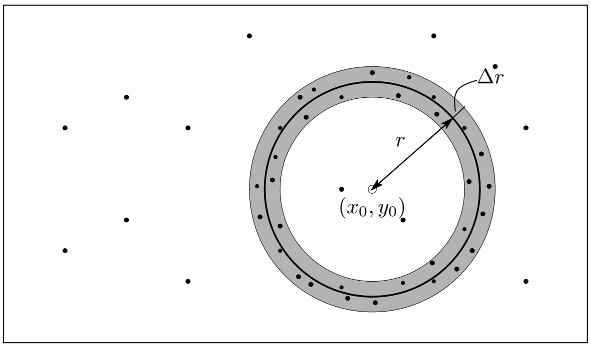

Fig. 1. Counting the points on a circle

is applied to the detection of high dimensional models such as the fundamental matrix [4][6].

This paper focuses on circles and adopts the RANSAC-based counting procedure. Thus, the number of the points on circle $f(\boldsymbol{r};\boldsymbol{a}) = (x-x_0)^2+(y-y_0)^2-r^2 = 0$, where $(x_0, y_0)^\top$ is the center and r is the radius, is counted by scanning the points satisfying the following

$$|(x - x_0)^2 + (y - y_0)^2 - r^2| \le \Delta r^2, \tag{3}$$

where Δr is a small constant for tolerating noise, as shown in Fig. 1.

III. State estimation Based on Sequential Monte Carlo filtering

Sequential Monte Carlo filters (SMC) [10] provide numerical estimates of marginal distribution $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$ up to time t recursively in time as follows:

$$\begin{aligned} &Prediction: \\ &p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1}) = \int p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1})d\boldsymbol{x}_{t-1}, \end{aligned} \tag{4}$$

$$\begin{aligned} &Filtering: \\ &p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}) = \frac{p(\boldsymbol{y}_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1})}{\int p(\boldsymbol{y}_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1})d\boldsymbol{x}_t}, \end{aligned} \tag{5}$$

where $\boldsymbol{x}_{0:t} \stackrel{\text{def}}{=} \{\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_t\}$ and $\boldsymbol{y}_{1:t} \stackrel{\text{def}}{=} \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_t\}$ are the signal and the observations up to time t, respectively. The algorithm, given an initial distribution $p(\boldsymbol{x}_0)$ and a transition equation $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})(= p(\boldsymbol{x}_t|\boldsymbol{x}_{0:t-1}))$, is generally described as follows [7][8]:

1. Initialization: $(t = 0)$
- For $i = 1, 2, \ldots, N$, sample $\boldsymbol{x}_0^{(i)} \sim p(\boldsymbol{x}_0)$ and set $t \leftarrow 1$.

2. Prediction:
- For $i = 1, 2, \ldots, N$, sample $\boldsymbol{x}_t^{(i)} \sim p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^{(i)})$

3. Filtering:
- For $i = 1, 2, \ldots, N$, evaluate the likelihoods
$$w_t^{(i)} = p(\boldsymbol{y}_t|\boldsymbol{x}_t^{(i)}).$$
- For $i = 1, 2, \ldots, N$, normalize the likelihoods
$$w_t^{(i)} \leftarrow \frac{w_t^{(i)}}{\sum_{k=1}^{N} w_t^{(k)}}.$$
- For $i = 1, 2, \ldots, N$, resample $\boldsymbol{x}_t^{(i)}$ according to the likelihoods.

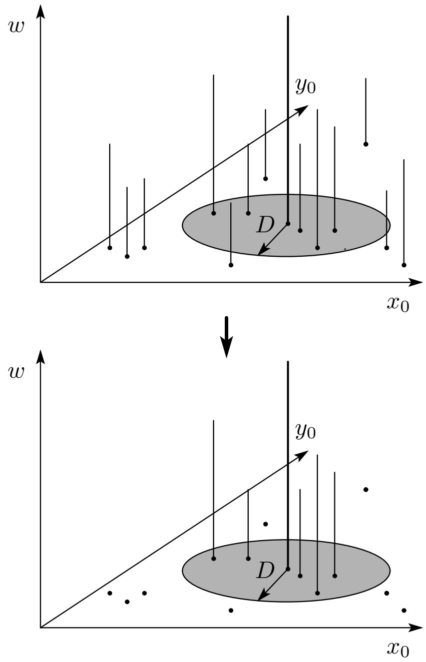

Fig. 2. Eliminating particles away from the mode for suppressing rapid diffusion

- Set $t \leftarrow t + 1$ and go to step 2.

Thus, the SMC filtering approximately provides the filtering distribution with a set of discrete finite points. This property enables the SMC filtering to handle nonlinear and non-Gaussian state-space models. Theoretically, the point-wise distribution asymptotically converges toward the true one if $N \to \infty$. However, due to limited computer resource, it is not practical to set the number of sample points, N, to be extremely large. To improve the performance of the SMC filtering without increasing the number of sample points, one should consider two issues:

- What kind of observation (image feature) is adopted?
- How is the likelihood evaluated?

IV. Temporal Peak Propagation for Circle Tracking by Sequential Monte Carlo Filtering

In the voting methods for detecting circles in Sec. II, peaks in the parameter space indicate the presence of circles in the image, i.e., the problem of detecting circles reduces to the peak detection. Hence, if the peak in an image of a sequence are propagated to the successive image, the searching of the peak in the image sequence can be efficiently performed. In the following, the algorithm of propagating the peak along time is proposed in the framework of the SMC filtering.

A. Generating Initial Distribution

Generating initial distribution $p(\boldsymbol{x}_0)$ is considered as the problem of detecting circles in the first frame of a give image sequence. Thus, after RANSAC or the Hough transform applies to the first frame, a set of particles $\{\boldsymbol{x}_0^{(i)}|i = 1, 2, \ldots, N\}$ is obtained by diffusing the state corresponding to the detected circle with adding system noise.

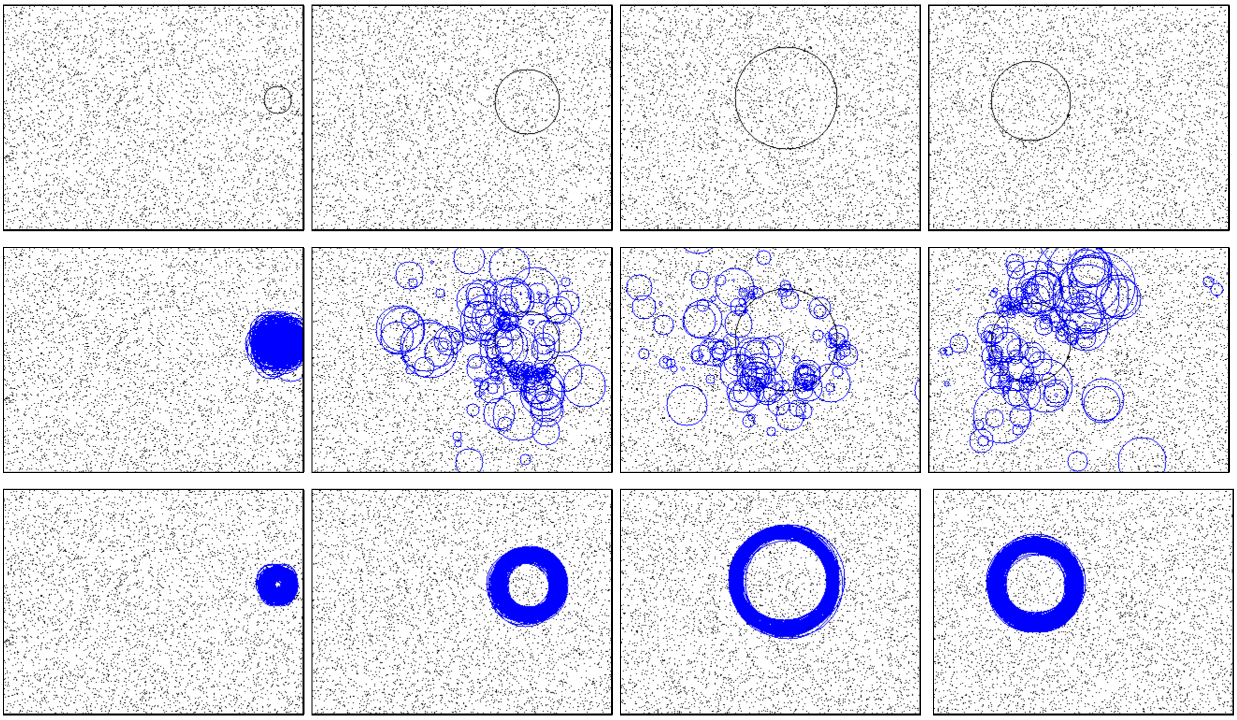

Fig. 3. (Top) examples of a sequence of cluttered images. (Middle) The tracking results without suppressing rapid diffusion. (Bottom) The tracking results with suppressing rapid diffusion.

B. One-Ahead Prediction: Peak Propagation

Assume that the center of the circle $(x_0(t), y_0(t))^\top$ smoothly moves in the sequence of images. Therefore a prior model of state transition

$$x_0(t+1) = 2x_0(t) - x_0(t-1), \quad (6)$$
$$y_0(t+1) = 2y_0(t) - y_0(t-1), \quad (7)$$

is adopted. As the state variables and the system noise,

$$\boldsymbol{x}_t = (x_0(t), y_0(t), r(t), x_0(t-1), y_0(t-1))^\top, \quad (8)$$
$$\boldsymbol{v} = (v_{x_0}, v_{y_0}, v_r)^\top, \quad (9)$$

are used. To sum up, the transition model is described by

$$\boldsymbol{x}_t = \boldsymbol{F}\boldsymbol{x}_{t-1} + \boldsymbol{G}\boldsymbol{v}, \quad (10)$$
$$\boldsymbol{F} = \begin{bmatrix} 2 & 0 & 0 & -1 & 0 \\ 0 & 2 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \boldsymbol{G} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad (11)$$

where $v_i, i = x_0, y_0, r$ are Gaussian white noise with mean 0 and variance $\sigma_i^2, i = x_0, y_0, r$ and are mutually independent. According to the model, the set of particles $\{\boldsymbol{x}_t^{(i)} | i = 1, 2, \ldots, N\}$ at time t which approximate the one-ahead prediction distribution $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ is generated from the set of particles $\{\boldsymbol{x}_{t-1}^{(i)} | i = 1, 2, \ldots, N\}$ at time $t-1$.

C. Filtering: Likelihood Evaluation Based on Voting

The likelihood of the approximates of the one-ahead prediction distribution, $\boldsymbol{x}_t^{(i)}, i = 1, 2, \ldots, N$, is defined by

$$w_t^{(i)} = I(\boldsymbol{H}\boldsymbol{x}_t^{(i)}) = \sum_k h(\boldsymbol{r}_k, \boldsymbol{H}\boldsymbol{x}_t^{(i)}), \quad (12)$$
$$h(\boldsymbol{r}_k, \boldsymbol{H}\boldsymbol{x}_t^{(i)}) = \begin{cases} 1 & \text{if } |f(\boldsymbol{r}_k; \boldsymbol{H}\boldsymbol{x}_t^{(i)})| \leq \Delta r^2, \\ 0 & \text{otherwise,} \end{cases} \quad (13)$$

where

$$\boldsymbol{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \quad (14)$$

and

$$f(\boldsymbol{r}_k; \boldsymbol{H}\boldsymbol{x}_t^{(i)}) = \left| \left(x_k - x_0^{(i)}(t)\right)^2 + \left(y_k - y_0^{(i)}(t)\right)^2 - r^2 \right|. \quad (15)$$

Note that the likelihood is normalized to $\sum_{i=1}^N I(\boldsymbol{H}\boldsymbol{x}_t^{(i)}) = 1$.

D. Resampling and suppressing rapid diffusion

The particles $\boldsymbol{x}_t^{(i)}, i = 1, 2, \ldots, N$, are resampled according to the likelihoods $w_t^{(i)}, i = 1, 2, \ldots, N$, to avoid the degeneracy for which most of the particles have almost zero weight after a few time steps.

However, tracking with resampling is likely to fail in clutter. For example, consider a tracking problem for cluttered binary images, as shown in Fig. 3 (top). Fig. 3 (middle) shows the results obtained by the voting-based tracking with resampling. The circles depicted in the images are the surviving particles after resampling at each time. From these results, the particles lose the circle and widely diffuse over the image with evolving

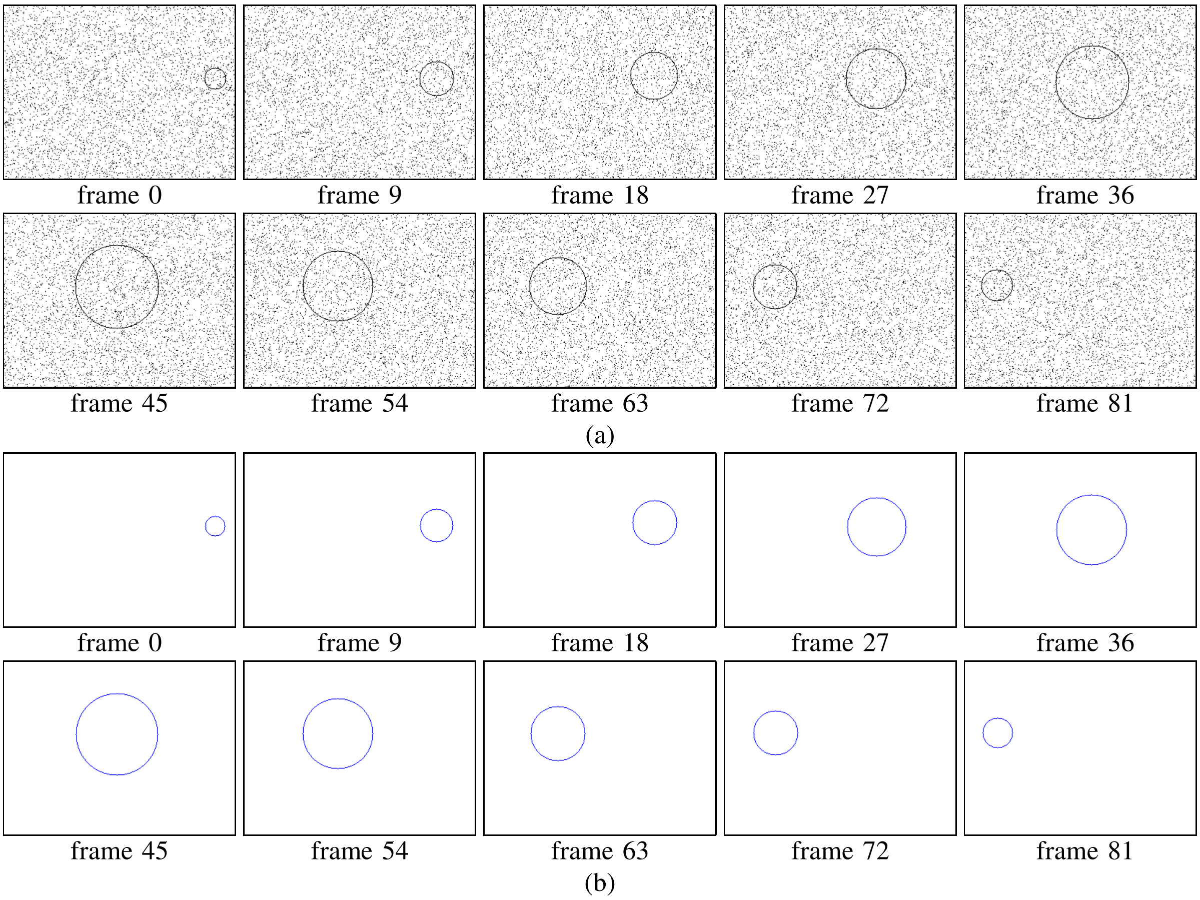

Fig. 4. (a) Examples of input images, (b) the tracking results by the voting-based tracking with suppressing rapid diffusion.

time. This phenomenon arises from the fact that, in cluttered images, the relatively high likelihood is given to the particles which correspond to background noise.

Thus, a procedure for suppressing the rapid diffusion is introduced. The procedure selects the mode of the filtering distribution $\{\boldsymbol{x}_t^{(i)} | i = 1, 2, \ldots, N\}$ before resampling:

$$m = \arg\max_i w_t^{(i)}, \tag{16}$$

and then set the weights of the particles which are distance D away from the mode to 0:

$$w_t^{(i)} = 0 \quad \text{if } \boldsymbol{x}_t^{(i)} \in S, \tag{17}$$

$$S = \left\{ \boldsymbol{x}_t^{(i)} \,|\, d\left(\boldsymbol{H}\boldsymbol{x}_t^{(i)}, \boldsymbol{H}\boldsymbol{x}_t^{(m)} \right) > D \right\}, \tag{18}$$

and finally perform resampling. Here, $d(\boldsymbol{x}, \boldsymbol{y})$ is the squared Mahalanobis distance defined by

$$d(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{(\boldsymbol{x} - \boldsymbol{y})^\top \boldsymbol{V}^{-1} (\boldsymbol{x} - \boldsymbol{y})}, \tag{19}$$

where

$$\boldsymbol{V} = \begin{bmatrix} \sigma_{x_0}^2 & 0 & 0 \\ 0 & \sigma_{y_0}^2 & 0 \\ 0 & 0 & \sigma_r^2 \end{bmatrix}. \tag{20}$$

In experiments of Sec. V, D is set to be 3. Since the particles such that $\left\{ \boldsymbol{x}_t^{(i)} | \boldsymbol{x}_t^{(i)} \in S \right\}$ have zero weight, they are eliminated after resampling and the other particles around the mode survive, as shown in Fig. 2. Fig. 3 (bottom) shows the result obtained by the voting-based tracking with suppressing rapid diffusion.

V. Experimental Evaluation for a Sequence of Cluttered Binary Images

To evaluate the performance of the proposed algorithm in clutter, the task of tracking a circle for two binary image sequences, as shown in Fig. 4(a) and Fig. 5(a) are made. Each image sequence consists of 90 frames with 320×240 pixels and each image is added to mutually independent 5000 random noises. In Fig. 4(a), a circle is used to evaluate the basic performance, and in Fig. 5(a), a circular arc, which lacks part of the circle, is used to evaluate the performance under a more realistic situation. The circle and the circular arc move in the same way; the time transition of the center $(x_0(t), y_0(t))^\top$ and the radius r is

$$x_0(t) = 290 - 3t + v, \tag{21}$$

$$y_0(t) = 100 + v, \tag{22}$$

$$r(t) = \begin{cases} r(t-1) + 1 + v, & \text{if } t < \frac{T}{2} \\ r(t-1) - 1 + v, & \text{otherwise} \end{cases}, \tag{23}$$

where v is a white noise generated from the Gaussian distribution with mean 0 and variance 3^2 (pixels) and constant T is the number of the frames.

Fig, 4(b) and Fig. 5(b) shows the tracking results by the proposed algorithm. In each experiment, the number of the

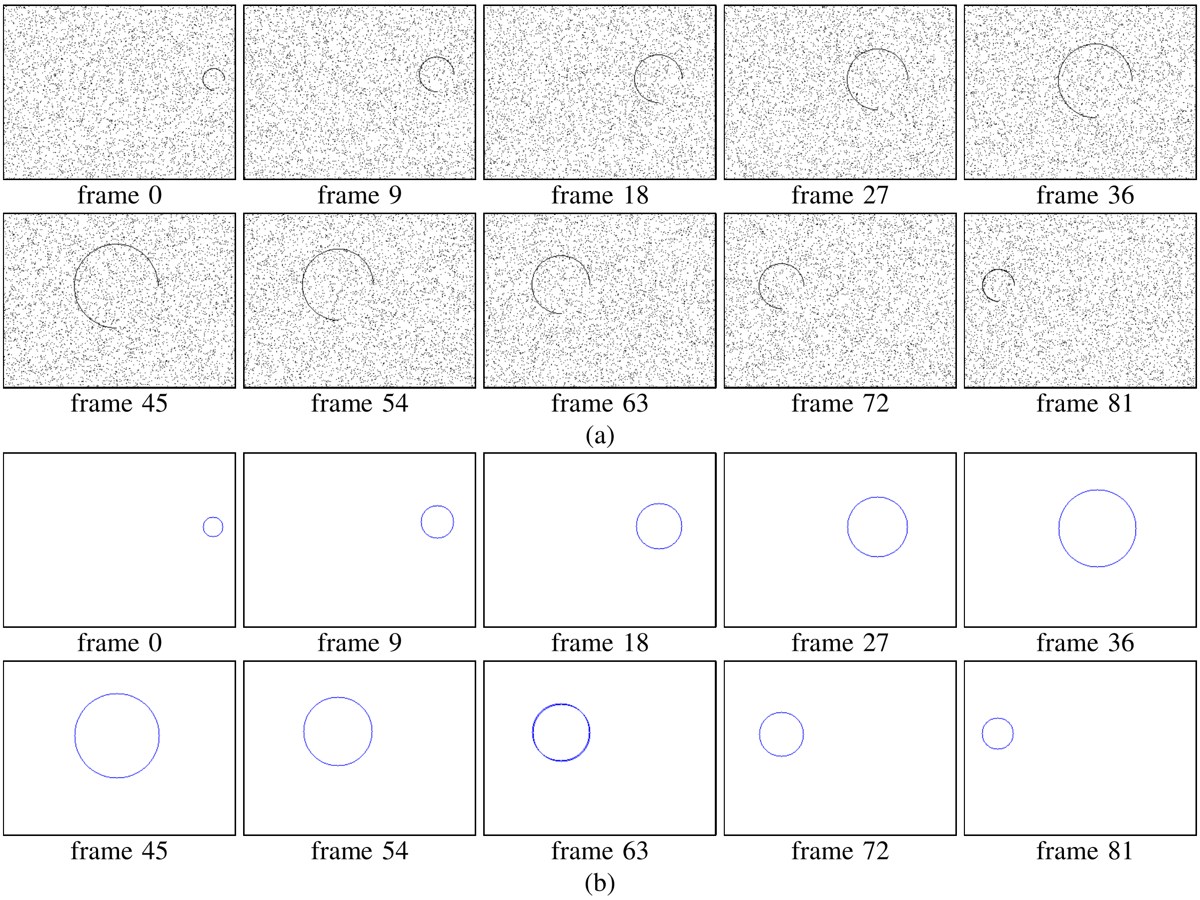

Fig. 5. (a) Examples of input images, (b) the tracking results by the voting-based tracking with suppressing rapid diffusion.

particles used is set to be $N = 500$ and the variance of the system noises is set to be $\sigma_i = 3^2(\text{pixels}), i = x_0, y_0, r$. The circles depicted in Fig, 4(b) and Fig. 5(b) correspond to the mode of the filtering distribution, which is the state with the highest weight, at each time, i.e., the mode is used as the estimate of the state at each time. In the framework of voting, it is reasonable to use the model as the estimate because the state with the highest weight (score) indicates the most likely candidate of the curve in the image. From these results in Fig, 4(b) and Fig. 5(b), the proposed algorithm successfully tracks the circle in clutter. The execution time per frame for each image sequence is 23.6 msec and 23.5 msec, respectively, on the average of 10 trials (except for input/output of images). The proposed method, therefore, works less than the standard video frame rate (30fps).

VI. Conclusions

This paper proposes a voting-based algorithm for visual tracking in noisy binary image sequences using a SMC filter. The proposed algorithm propagates votes, which indicate the candidates of curves, obtained in an image of a given sequence to the successive image. Although the voting-based algorithms such as the Hough transform and RANSAC, in general, are time-consuming, this propagation over time provides an efficient search procedure for finding curves in the framework of voting. Thus, two combinations of the voting and the SMC filtering enable us to track curves in clutter robustly and efficiently. Furthermore, a particle elimination procedure is introduced to suppress rapid diffusion of particles over time, which phenomenon is likely to happen in clutter. As a result, the elimination procedure prevents the variance of the filtering distribution from increasing over time.

Robust and efficient algorithms in image processing and computer vision are required in academic and industry because of the spread of computers and digital cameras. The proposed algorithm gives a basis of such image and vision technologies.

Acknowledgements

This work was supported by the Grant-in-Aid for Young Scientists B (No.16700169).

References

[1] R. O. Duda and P. E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *Comm. ACM*, vol. 15(1), pp. 11–15, Jan. 1972.

[2] M. A. Fischer and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Comm. ACM*, vol. 24(6), pp. 381–395, June 1981.

[3] H. Kalviainen, O. Hirvonen, L. Xu, and E. Oja, "Probabilistic and non-probabilistic Hough transforms: Overview and comparisons," *Image and Vision Computing*, vol. 13(4), pp. 239–252, 1995.

[4] Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong, "A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry", *Artificial Intelligence*, vol. 78(1), pp. 87–119. Oct. 1995.

[5] O. Chum and J. Matas, "Randomized RANSAC with $T_{d,d}$ test," *Proc. the British Machine Vision Conference*, vol. 2, pp. 448-457, Sep. 2002.

[6] P.H.S. Torr, and C. Davidson, "IMPSAC: synthesis of importance sampling and random sample consensus", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25(3), pp. 354–364, March 2003.

[7] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation", *IEE Proc.–F*, vol. 140(2), pp. 107–113, April 1993.

[8] G. Kitagawa, Monte Carlo filter and smoother for non-Gaussian nonlinear state space models, *J. Comput. Graph. Stat.*, Vol.5, No.1, pp. 1–25, 1996.

[9] M. Isard and A. Black, "Condensation – Conditional density propagation for visual tracking," *Int. J. Computer Vision*, vol. 29(1), pp.5–28, 1998.

[10] A. Doucet, N. de Freitas, and N. J. Gordon, "*Sequential Monte Carlo Methods in Practice*", Springer-Verlag, May 2001.