International Journal of Multimedia and Ubiquitous Engineering
Vol.10, No.11 (2015), pp.265-276
http://dx.doi.org/10.14257/ijmue.2015.10.11.25

On the Accuracy of Particle Filter-Based Object Tracking

Dewa Made Wiharta[1,2], Wirawan[1] and Gamantyo Hendrantoro[1]

[1]Department of Electrical Engineering,
Institut Teknologi Sepuluh Nopember (ITS), Surabaya, Indonesia
[2]Departement of Electrical Engineering,
Universitas Udayana, Denpasar, Indonesia
wiharta@unud.ac.id, wirawan@its.ac.id, gamantyo@ee.its.ac.id

Abstract

In particle filter framework, there are many parameters need to be set to have the object tracking work best. On the other hand, the success of an object tracking is also determined by the characteristics of the video being used. Different videos might require different values of parameters to obtain best performance. This paper investigates thoroughly the effect of parameters in object tracking within particle filter framework for videos with different characteristics. We also propose a new method in selecting particle number and matrix value that used in propagation model. This work finds best parameters for video with particular characteristics and contributes in providing guideline of parameter setting in particle filter-based object tracking.

***Keywords**: object tracking, particle filter, tracking performance, color histogram*

1. Introduction

Object tracking is a problem of estimating object position in image sequences. Many researchers put their interest in this field since object tracking is related to issues about automated surveillance, motion-based recognition, human-computer interaction, and automated video analysis. Many approaches for object tracking have been proposed. In this paper we apply particle filter as framework in object tracking task as it is proven to be robust and work well in non-Gaussian environments [1-2].

Particle filtering is a sequential Monte Carlo method that computes the posterior probability density function of a state space recursively by means of Bayesian equation. This sequential Monte Carlo (SMC) approach is known variously as bootstrap filtering, the condensation algorithm, particle filter, interacting particle approximation, and survival of the fittest [3-4]. The state space model is changing in time and the information about the state is gathered through a noisy observations. In general, state of the system is changed according to equation [5]:

$$\mathrm{x}_k = f(\mathrm{x}_{k-1}, \mathrm{v}_{k-1}) \tag{1}$$

Where x_k is vector that define the state of the system at time k, v_k is state of noise vector, and f_k is non-linear time-dependent function to describe evolution of state vector. Information of x_k is collected from noisy observation by the following equation:

$$\mathrm{z}_k = h(\mathrm{x}_k, \mathrm{n}_k) \tag{2}$$

Particle filter has been implemented in many tracking problems. The aim of tracking is to estimate x_k recursively from observation (2). The basic idea of particle

ISSN: 1975-0080 IJMUE

filter is to approximate the posterior probability density by a set of samples S, where each sample (particle) is weighted by ω_k.

$$S = \{x^i_{0:k}, \omega^i_k\}^N_{i=1} \tag{3}$$

Weight ω_k is normalized as:

$$\sum_i \omega^i_k = 1 \tag{4}$$

Sample S gives a random measure of posterior pdf $p(x_{0:k}|z_{1:k})$ where $x_{0:k} = \{x_j, j = 0, \ldots, k\}$ is the set of all state up to time k [5]:

The set S is propagated through a first order dynamic model :

$$S_t = \mathbf{A}\, s_{t-1} + \mathbf{B}\, w_{t-1} \tag{5}$$

Where A is deterministic component in form of 4 × 4 matrix and **B** is a constant matrix to regulates particles distribution and w is multivariate Gaussian random variable.

Performances of object tracking is determined by parameters such as number of particles N, variance of Gaussian random noise σ, number of histogram bins, and propagation constant **B**. Increasing number of particles N will enhance the accuracy of the tracking with the cost of computation time. The values of σ and **B** regulate the spreading of particles which also affects the tracking performance.

Object representation also plays important role in object tracking. The most common feature used by many researchers to represent the object is color histogram [1,6-9]. The color histogram can be built by the use of three-dimensional color, like RGB, by dividing color values into bins with uniform interval. Larger value of bin means smaller interval which gives better feature, but increases the computation time.

Different researchers use different bin configurations and numbers of particles without explaining the reasons and the impacts they have on the accuracy of tracking. They also did not explicitly mention the noise and the spreading of particles. For example, ref. [1] uses bin configuration of 8 × 8 × 8 of RGB color space with N = 75. Ref. [9] compares the use of 75 and 500 particles for 8 × 8 × 8 bin configuration and ref. [10] uses bin 10 × 10 × 10 for HSV color space. 200 particles is used by [11] with 10 × 10 × 10 RGB color space.

There is no paper that studies thoroughly the impact of number of particles, bin configurations and other parameters such as variance of Gaussian noise and particle spreading to the accuracy of tracking. In this paper, we investigate the values of those parameters by means of rigorous simulations in particle filter-based object tracking to achieve best performance in term of tracking accuracy. We also propose a new method to find number of particle and the constant matrix **B**. The number of particle is set proportional to target object resolution (number of pixel within object region), and **B** is set relative to object size.

Another contribution of this paper is finding optimum parameters for particle filter-based object tracking. The evaluation method is performed by a thorough test of varying parameters for various videos with different characteristics.

The paper is organized as follows. In Section 2, we briefly review particle filter in the context of object tracking. The method of setting parameter values is described in Section 3. Section 4 presents the test videos and the experimental results, followed by conclusion in Section 5.

2. Particle Filter

In particle filter, state of an object being tracked is defined as vector x_k and the observation up to time-k is defined as vector z_k. Particle filter often used when posterior density $p(x|z)$ and observation density $p(z|x)$ are non-Gaussian. In this research, we apply particle filter in the context of object tracking with the use of color space as object representation.

Firstly, color feature space is chosen to provide the characteristics of the object. Then, the target is represented in probability density function (pdf), noted by q, in the feature space. The target model is defined to be located in the spatial location (0,0). In the next frame, target candidate, defined at location y, is characterized by pdf p(y). The object location then estimated by measuring the different between q and p(y).

2.1. Target Model

In order to reduce computation cost, we use m bins color histogram to represent target model. The target model is defined as:

$$q = \{q_u\}_{u=1....m} \tag{6}$$

Where m is the number of bins, and q is the target model.

The probability of $u = 1, \ldots, m$ in target model is computed by [12]

$$q_u = C\sum_{i=1}^{n}(k\|x_i^*\|^2)\delta\left[b(x_i^*) - u\right] \tag{7}$$

Where δ is Kronecker delta function and k is a monotonic decreasing kernel that assign smaller weight to pixels that farther from object center.

2.2. Target Candidates

Let $\{x_i^*\}_{i=1,\ldots,n}$ be the pixel located in a region of target model. Color histogram is produced by function $h(x_i)$ that assigns the color at location x_i to the corresponding bin, i.e. the index $h(x_i^*)$ of its bin in the quantized feature space. The target candidate $p_u(y)$ is computed as [12] :

$$p_u(y) = C\sum_{i=1}^{n} k(\|y - x_i\|)\delta[h(x_i) - u] \tag{9}$$

Where δ is the Kronecker delta function. The normalization constant C is derived such that $\sum_{u=1}^{m} p_u(y) = 1$

2.3. Bhattacharyya Distance

The state estimate is updated in each iteration using new observation. Since we use color histogram as feature, we require a measure of similarity based on the color distribution. Similarity measurement of two distribution p_u and q_u is provided by Bhattacharyya coefficient [13]:

$$\rho[p(y), q] = \sum_{u=1}^{m}\sqrt{p_u(y)q_u} \tag{10}$$

The value of ρ defines distance between p and q. Larger ρ means that both distributions are more similar. For normalized distribution, $\rho = 1$ is achieved by two identical distributions.The distance between two distributions is defined by

$$d(y) = \sqrt{1 - \rho[p(y), q]} \tag{11}$$

Known as Bhattacharyya distance.

3. Object Tracking in Particle Filter Framework

3.1. The Algorithm

We use a generic particle filter with resampling [5]. The particles distribution is regulated by the motion model and the measurement noise as variance of Gaussian distribution (σ). Setting too small value of σ will lead to degeneracy problems, while setting σ too large spreads particles widely that tends to reduce tracking accuracy. We modify the original motion model in [1,14] by adding a constant matrix **B** to further regulates particle distribution.

The Algorithm for object tracking is presented below:

1. Initialize by defining object to be tracked
2. Calculate the distribution of the target model q

$$q_u = C \sum_{i=1}^{n} (k \| x_i^* \|^2) \delta \left[b(x_i^*) - u \right]$$

3. Propagate N sample set using motion model S_k.

$$\mathbf{S}_k = \mathbf{A}s_{k-1} + \mathbf{B}w_{k-1}$$

4. Calculate the distribution for each sample of the set using

$$p_u(y) = C_h \sum_{i=1}^{n} k(\left\| y - x_i \right\|^2) \delta [b(x_i) - u]$$

5. Calculate the Bhattacharyya coefficient for each sample

$$\rho[p, q] = \int \sqrt{p_u q_u}\, du$$

6. Compute weight

$$\omega = \frac{1}{N} \left(\frac{1}{\sqrt{2\pi}\sigma}\right) e^{(\rho - 1)/(2\sigma^2)}$$

7. Resample
8. Estimate the mean state

$$E[S_k] = \sum_1^N \omega^i s_k^i$$

3.2. Environment and the Parameter Setting

The experiments run on 32-bit CPU 2.4 GHz. We use RGB color space as feature space. Tracking performances are qualified by Root Mean Square Error (RMSE).

a. Number of Particles

The first experiment is performed by varying number of particles N to be used on tracking. This number is set proportional to target object resolution R (number of pixel within object region). For example, for target object of a circle with radius 17 pixels, the resolution R is computed as area of a circle = 907 pixels. N is than varied from $R/10$, $R/8$, $R/4$, $R/2$, R, and $2R$.

b. Bin

Color histogram represents the number of pixels that have colors in each bin that span the image's color. The bin configurations that we test are 4 × 4 × 4; 6 × 6 × 6; 8 × 8 × 8; 12 × 12 × 12 and 16 × 16 × 16.

c. Variance of Gaussian Noise σ

The variance of Gaussian measurement noise is varied to the value of 0.05, 0.1, 0.15, 0.2 and 0.25.

d. The Constant Matrix B

The dimension of propagation matrix **B** is 4 × 4. The values are set as: **B** = [1 d_1 0 0; 0 1 0 0 ; 0 0 1 d_2 ; 0 0 0 1]; where d_1 and d_2 are varied relative to the object dimension d_x and d_y as shown in Figure 1. The values of d_1 and d_2 start from 1/8, ¼. ½, 1, 1.5 dan 2 time of object dimensions d_x and d_y, and we denote them as B_1, B_2, B_3, B_4, B_5, and B_6 respectively.

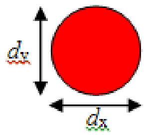


Figure 1. The Value of B is Relative to the Size of the Object

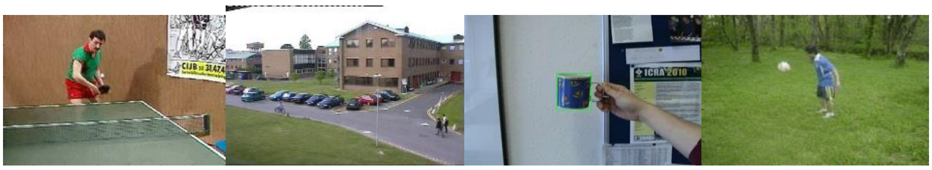
4. The Experiments

4.1. The Test Videos

We perform particle filter-based object tracking using several videos that have been used by various literatures. The first video is a slow movement ping-pong player video used in [15-16]. Car video from PETS database (http://ftp.pets.rdg.ac.uk/PETS2000/) used in [17-18] and cup video [19] are another videos of slow moving objects. The differences are that there are similar objects in the car video and scaling and moving background for cup video.

For object with changing speed and direction, we track ball in the juggling ball video from http://fcl.uncc.edu/ nhnguye1/balltracking.html and the ping pong ball video used in various literatures [7,11,16,20].

Video of moving cam, rotation, and direction changes from Bonn Benchmark on Tracking (BoBot) (http://www.iai.uni-bonn.de/~kleind/tracking/) is used as the sixth test video. This video is used in [19,21]. Tracking object with occlusion is performed by video from trictrac (http://www.multitel.be/trictrac/) [22] as the seventh video. The eighth video is of soccer player with deformable and dynamic movement. Figure 2 displays the test videos.



(a) Ping-Pong Player (b) Car (c) Cup (d) Juggling Ball

(e) Ping-Pong ball (f) BoBot Ball (g) Trictrac Ball (h) Soccer

Figure 2. Videos Used in the Experiments

4.2. Results

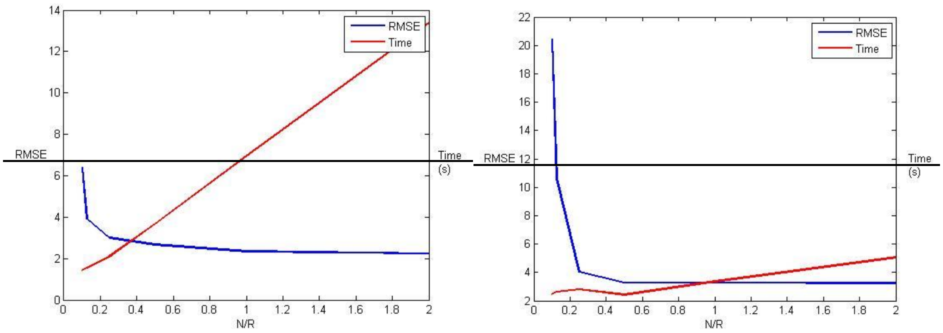
We perform the experiments on the test videos and find the best parameters that deliver best results in term of accuracy (RMSE) and computation time. Table 1 summarizes the results.

Table 1. Tracking Results

Video	object	N	Bin	σ	B	RMSE	Time (s)
Ping-pong player	head	R/4	8 8 8	0.05	$d_x/4$; $d_y/4$	1.6062	3.3707
Juggling ball	ball	R	4 4 4	0.15	d_x ; d_y	1.8194	3.1229
Ping-pong ball	ball	R/2	6 6 6	0.2	d_x ; d_y	2.9144	2.2437
BoBoT Ball	ball	R	4 4 4	0.2	$d_x/2$; $d_y/2$	3.1209	132.174
Cup	cup	R/4	6 6 6	0.05	$d_x/4$; $d_y/4$	3.7875	82.4762
Trictrac	ball	R	8 8 8	0.2	$d_x/2$; $d_y/2$	5.6141	66.8452
Soccer	player(white)	R	6 6 6	0.2	$d_x/2$; $d_y/2$	3.1777	18.2103
Car	car	R	12 12 12	0.2	$d_x/2$; $d_y/2$	4.0905	35.8206

a. Number of Particle N

Video with different characteristics require different parameter values to achieve best performance. Increasing number of particles surely increases accuracy and computation time as well. For videos with slow object movement, we find that small number of particles (N), as much as a quarter of object resolution, already deliver good tracking accuracy. Further increase of N does not give significant improvement, while the computation cost is greatly increased. Therefore, there is a tradeoff between accuracy and computation time. Examples of tradeoff are shown in Figure 3.



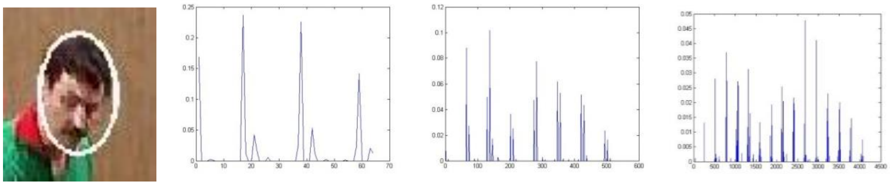
(a) (b)

Figure 3. Trade-Off Between Accuracy and Computation Cost (a) Ping-Pong Player ; (b) Ping-pong Ball Video

Other videos with faster moving objects such as juggling ball and car video require larger number of particles, relative to object dimension (See Table 1).

b. Bin

In investigating the effect of bin to tracking accuracy, we perform the experiments by varying number of bin. Bin 4 4 4 creates histogram of 4 × 4 × 4 = 64 dimensions and bin 16 16 16 creates histogram of 16 × 16 × 16 = 4096 dimensions. Larger number of bin yields histogram with smaller interval. Figure4 shows histograms of a target object and its histograms in various bin configurations.



(a) (b) (c) (d)

Figure 4. Histogram of a Target Object. (a) The Target Object; (b) Histogram of Bin 4 4 4 ; (c) Histogram of Bin 8 8 8; (d) Histogram of Bin 16 16 16

We may assume that larger bin number gives better performance in the cost of computation time. However, this is not true for some videos. Objects with one dominant color, such as the white ping-pong ball, performs best with $4 \times 4 \times 4$ bin configuration, while objects with wider color span, such as player head in ping-pong player and ball from trictrac video requires larger bin configuration to achieve accurate tracking.

c. Variance of Gaussian Noise

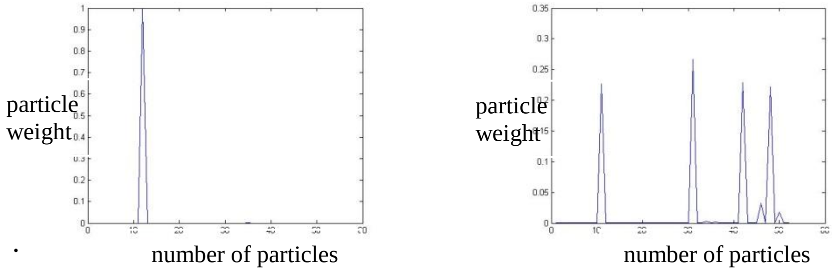
Varying σ is affecting the accuracy of tracking, not the computation time. Setting small values of measurement or variance noise σ might lead to degeneracy problem [5]. This is a common problem with particle filters. After a few iterations, a large number of particles will have negligible weight. It is impossible to avoid the degeneracy phenomenon since the variance of the importance weights increase over time [23]. Resampling strategy is used to reduce the effects of degeneracy. The degeneracy is measured by an effective sample size N_{eff} [24]:

$$N_{eff} = \frac{N}{1+\mathrm{Var}(\omega_k^{*i})} \tag{12}$$

This measure cannot be evaluated exactly, but can be estimated as [5]:

$$\widehat{N_{eff}} = \frac{1}{\sum_{i=1}^{N}(\omega_k^i)^2} \tag{13}$$

In our experiments, we find that the value of σ determines the number of particle having nonzero weight. Small value of σ causes only a few particles that have nonzero weight and large value of σ tends to spread particles in a large area. In juggling ball video, the value of $\sigma = 0.05$ produces only two out of 52 particles that have nonzero weight. One of them weighted almost one, which results in N_{eff} of 1.0003. This will lead to poor result since only too few particles contribute in the tracking. Increasing σ distributes weight more equally to the particles with better tracking performance. Figure5 displays distribution of nonzero weight particles for small and large σ.

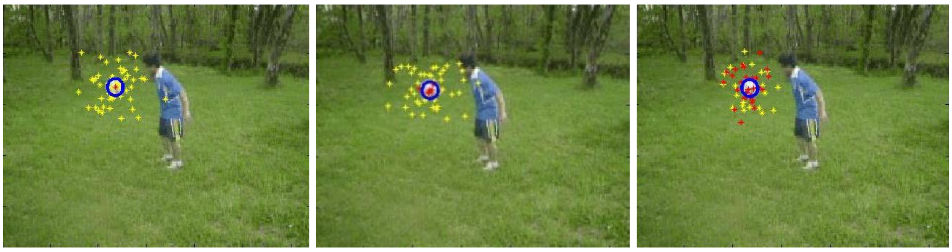


(a) (b)

Figure 5. Nonzero Weight Particles. (a) Small Value of σ Reduces Nonzero Weight Particles; (b). A More Distributed Weight with Larger σ

As described earlier, increasing σ generates more nonzero weight particles and yields in larger N_{eff}. But too large σ will spread particles (after resampling) in a larger area, and only a small portion of them fall in the tracked object, as shown in Figure 6. Small number of particles that fall on the tracked object means only few particles that contribute to tracking estimate, which reduce tracking performance. The distributions of particles are displayed in red and yellow star. Yellow stars are the initial particles and red stars are particles after resampling. (a). σ = 0.05 generates only two nonzero particles after resampling; (b). σ = 0.2 distributes weights more equally among particles; (c). σ = 0.4 spreads particles widely that reduce tracking accuracy.

In video where objects move slowly between frames, we find that small values of σ perform best as in ping-pong player and cup video. Larger value of σ is required for faster moving objects, such as the juggling ball video.



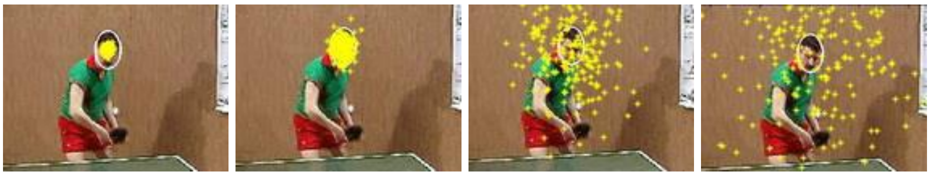
(a) σ = 0.05 (b) σ = 0.2 (c) σ = 0.4

Figure 6. Particles Distribution

d. The Constant Matrix B

The constant matrix **B** defines the spreading of particles prior resampling. Figure7 displays how particles spread in various value of **B** for video of ping-pong player.

Small value of **B** produces high effective particles, i.e. particles that fall on the object to contribute in the tracking estimate. B_1 has 90 % of effective particles with equal weights. Increasing **B** helps in spreading particles in a wider area, but over spreading tends to reduce particles that fall on the object, as shown in Figure7(d), which results in poor tracking performance.



(a) B_1 (b) B_3 (c) B_5 (d) B_6

Figure 7. Particles Distribution. Small Value of B Makes Particles Clustered in a Small Area Within Tracked Object. Increasing B Spreads Particles More Widely

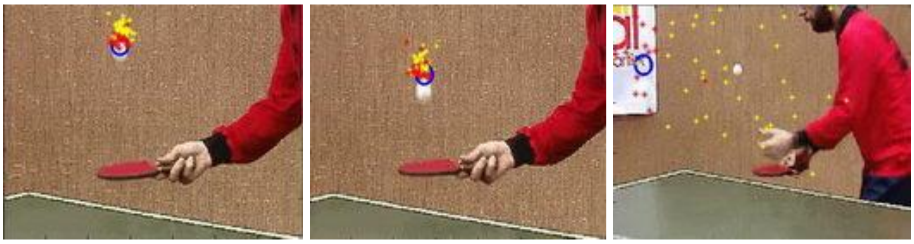
In the video where object displacement between successive frames is small, the tracking still performs well though the value of **B** is set small. But this is not true for object that move fast and change direction. The juggling ball video in Figure8 illustrates this chase. In this figure, the red circle is the tracking estimate and the yellow marks are the particles distribution

The object (ball) displacement before and after hitting player shoe between frame 50 and 51, and between frame 65 and 66, are quite large compared to other frames. Small value of **B** values fails to track object as in Figure8 (b) where most of particles follow the player shoe. In this case, we need a larger value of **B** to ensure that most particles follow the object of interest as in Figure8(c). Few particles that follow the shoe will be replaced with new particles after resampling,

(a) (b) (c)

Figure 8. Tracking Performances in Changing Speed and Direction

In tracking ping-pong ball, small **B** fails in tracking object with fast movement and abrupt direction change. Figure9 illustrates these conditions. The yellow and red marks are particle distribution before and after resampling respectively.



(a) frame 7 (b) frame 9 (c) frame 49

Figure 9. Particle Distribution for Ping-pong Ball Video. The Blue Circle is the Tracking Estimate

Small value of **B** makes particles to cluster within the object, reducing the diversity between particles and the probability to capture object with fast movement.

The tracking is failed as shown in Figure9 (a) and (b). Large value of **B** in Figure9 (c) spreads particles widely, which also result in failure of tracking. The proper value of **B** yields a good tracking result as shown in Figure10.

(a) frame 8 (b) frame 11 (c) frame 12

Figure 10. Setting Proper Value of B Leads to Accurate Tracking Performance

Large value of **B** is also useful when dealing object with occlusion. The trictrac ball video in Figure11 explains this case.

(a) (b) (c)

Figure 11. Tracking Ball in Trictrac Video. (a) The Ball is Occluded in Frame 751; (b) Small B in Frame 753; (c) Larger B Tracks Object After Being Occluded.

In frame 751, only small portion of the tracked ball is visible. Small **B** does not have particles fall in the object, and tracking is failed. Larger value of **B** distributes particles widely to ensure that some particles fall in the object to continue the tracking.

5. Discussions and Conclusions

We have presented comprehensively the effects of parameter values in particle filter-based object tracking. Our experiments show that video characteristics should be taken into account when setting values of tracking parameters.

Number of particles defines number of samples in tracking. Theoretically, larger number of particles approaches optimal Bayesian estimate. In this experiment, we find that particles as many as half of object resolution has already given good accuracy, and increasing particles does not significantly increase performance while computation time is greatly increased.

The interval of color histogram is defined by the number of bins. More bins create denser histogram. Different video requires different bin configuration for best performance. Lower bin configuration suits better for object dominated by single color. Larger bin configuration is required when the color span of the object is wide.

The accuracy of tracking is also determined by the spreading of particles, which is regulated by variance noise σ and constant matrix **B**. Setting these values depend on the characteristics of the video to achieve best tracking performance. Small value of **B** works for object with small displacement between frames (slow movement object). Larger **B** is required when the object movement is fast, change direction, or being occluded to ensure that some particles fall in the object.

The value of σ determines the number of particles having nonzero weight. Small value of σ causes particles to cluster in a small area with only a few particles having nonzero weight, and large value of σ generates more nonzero weight particles which tends to spread in a large area. Either one might result in poor accuracy or even tracking failure.

In our experiments with eight different videos, the best values of σ range from 0.05 to 0.2, depends from movement speed of the object. In video where objects move slowly between frames, we find that small values of σ perform best as in ping-pong player and cup video. Larger value of σ is required for faster moving objects, such as the juggling ball video.

Acknowledgements

This work is supported by Indonesian Ministry of Education and Culture through the BPPS scholarship.

References

[1] K. Nummiaro, E. K. Meier and L. V. Gool, "An adaptive color-based particle filter", Image Vis. Comput., vol. 21, no. 1, **(2003)**, pp. 99–110.
[2] A. Yilmaz, O. Javed and M. Shah, "Object tracking: A survey," Acm Comput. Surv. CSUR, vol. 38, no. 4, **(2006)**, pp. 13.
[3] N. J. Gordon, D. J. Salmond and A. F. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation", in IEE Proceedings F (Radar and Signal Processing), vol. 140, **(1993)**, pp. 107–113.
[4] M. Isard and A. Blake, "Condensation—conditional density propagation for visual tracking", Int. J. Comput. Vis., vol. 29, no. 1, **(1998)**, pp. 5–28.
[5] M. Arulampalam, S. Maskell, N. Gordon and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking", Signal Process. IEEE Trans. On, vol. 50, no. 2, **(2002)**, pp. 174–188.
[6] P. A. Brasnett, L. Mihaylova, N. Canagarajah and D. Bull, "Particle filtering with multiple cues for object tracking in video sequences", in Electronic Imaging, vol. 2005, **(2005)**, pp. 430–441.
[7] D. Comaniciu, V. Ramesh and P. Meer, "Kernel-based object tracking", IEEE Trans. Pattern Anal. Mach. Intell., vol. 25, **(2003)**, pp. 564–577.
[8] I. Laptev, "Improving object detection with boosted histograms", Image Vis. Comput., vol. 27, no. 5, **(2009)**, pp. 535–544.
[9] C. Teulière, E. Marchand and L. Eck, "A combination of particle filtering and deterministic approaches for multiple kernel tracking", in IEEE Int. Conf. on Robotics and Automation, ICRA'09, **(2009)**, pp. 3948–3954.
[10] W. Lu, K. Okuma and J. Little, "Tracking and recognizing actions of multiple hockey players using the boosted particle filter", Image Vis. Comput., **(2009)**, pp. 189–205.
[11] A. Naeem, T. Pridmore and S. Mills, "Managing Particle Spread via Hybrid Particle Filter/Kernel Mean Shift Tracking", presented at the British Machine Vision Conference, **(2007)**.
[12] D. Comaniciu, V. Ramesh and P. Meer, "Real-Time tracking of non-rigid objects using mean shift", presented at the IEEE CVPR, **(2000)**, pp. 142–149.
[13] T. Kailath, "The divergence and Bhattacharyya distance measures in signal selection", Commun. Technol. IEEE Trans. On, vol. 15, no. 1, **(1967)**, pp. 52–60.
[14] W. Li, J. Cao and D. Wu, "Multi-feature Fusion Tracking Based on A New Particle Filter", J. Comput., vol. 7, no. 12, **(2012)**, pp. 2939–2947.
[15] J. Ning, L. Zhang, D. Zhang and C. Wu, "Robust Object Tracking Using Joint Color-Texture Histogram", Int. J. Pattern Recognit. Artif. Intell., vol. 23, no. 7, **(2009)**, pp. 1245–1263.
[16] J. Ning, L. Zhang, D. Zhang and C. Wu, "Robust mean-shift tracking with corrected background-weighted histogram", IET Comput. Vis., vol. 6, no. 1, January **(2012)**, pp. 62–69.
[17] M. Heikkila and M. Pietikainen, "A texture-based method for modeling the background and detecting moving objects", IEEE Trans. Pattern Anal. Mach. Intell., vol. 28, no. 4, **(2006)**, pp. 657–662.
[18] F. Bousetouane, L. Dib and H. Snoussi, "Improved mean shift integrating texture and color features for robust real time object tracking", Vis. Comput., vol. 29, no. 3, February **(2012)**, pp. 155–170.

[19] D. Klein, D. Schulz, S. Frintrop and A. Cremers, "Adaptive real-time video-tracking for arbitrary objects", presented at the International Conference on Intelligent Robots and Systems, **(2010)**, pp. 772–777.
[20] E. Maggio and A. Cavallaro, "Video tracking: theory and practice", John Wiley & Sons, **(2011)**.
[21] G. M. García, D. A. Klein, J. Stückler, S. Frintrop and A. B. Cremers, "Adaptive Multi-cue 3D Tracking of Arbitrary Objects", in Pattern Recognition, A. Pinz, T. Pock, H. Bischof, and F. Leberl, Eds. Springer Berlin Heidelberg, **(2012)**, pp. 357–366.
[22] Desurmont J.-B. Hayet, C. Machy, J.-F. Delaigle and J.-F. Macq, "On the performance evaluation of tracking systems using multiple pan-tilt-zoom cameras", in Videometrics IX, **(2007)**.
[23] A. Doucet, S. Godsill and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering", Stat. Comput., vol. 10, no. 3, **(2000)**, pp. 197–208.
[24] A. Doucet and A. M. Johansen, "A tutorial on particle filtering and smoothing: Fifteen years later", Handb. Nonlinear Filter., vol. 12, **(2009)**, pp. 656–704.

Authors

Dewa Made Wiharta, is a lecturer in Universitas Udayana, Denpasar, Bali, Indonesia, received the M. Eng. degree in telecommunication information system from Universitas Gadjah Mada, Jogjakarta, Indonesia, 2002. Currently he is pursuing his doctorate degree at the Department of Electrical Engineering, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia. His research interests include multimedia signal processing and telecommunication engineering.

Wirawan, received the DEA degree in informatic and signal processing from Ecole Superieure en Sciences Informatiques Sophia-Antipolis, France, 1996 and Doctor degree from Telecom Paris Tech, Paris, France, 2003, in the same field. He is currently a lecturer and head of multimedia communication lab in the Department of Electrical Engineering, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia. His research interests include wireless multimedia sensor networks and multimedia signal processing.

Gamantyo Hendrantoro, received the M. Eng. and PhD degree in telecommunication engineering from Electrical Engineering, Carleton University, Canada, 1997 and 2001. He is a Professor and head of antenna and propagation lab in the Department of Electrical Engineering, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia. His research interests include wireless digital communication with emphasis on radio channel characterization and modeling.

