

OPTIMIZED PARTICLES FOR 3D TRACKING

HUIYING CHEN

*Department of Industrial and Systems Engineering,
The Hong Kong Polytechnic University, Hung Hom,
Kowloon, Hong Kong, P. R. China
velvet.chen@polyu.edu.hk*

YOUFU LI

*Department of Manufacturing Engineering
and Engineering Management,
City University of Hong Kong,
83 Tat Chee Avenue, Kowloon, Hong Kong, P. R. China
meyfli@cityu.edu.hk*

Received 21 March 2011

Accepted 7 October 2011

3D visual tracking is useful for many applications. In this paper, we propose two different ways for different system configurations to optimize particle filter for enhancing 3D tracking performances. On the one hand, a new data fusion method is proposed to obtain the optimal importance density function for active vision systems. With this method, the importance density function in particle filter can be modified to represent posterior states by particle crowds in a better way. Thus, it makes the tracking system more robust to noise and outliers. On the other hand, we develop a method for reconfigurable vision systems to maximize the effective sampling size in particle filter, which sequentially helps to solve the degeneracy problem and minimize the tracking error. Simulation and experimental results verified the effectiveness of the proposed method.

Keywords: 3D tracking; particle filter; importance density function.

1. Introduction

Three-dimensional (3D) tracking deals with continuous 3D state estimation and update of moving objects.¹ The task of 3D tracking is of paramount importance for many applications and has been considered from widely different perspectives of various theoretical backgrounds and interests. As one of the state-space estimation problems, 3D tracking can be modeled with the aid of parametric models. However, due to varying degrees of uncertainty inherent in system modeling and complexity of system noise, visual system is often subject to elements of non-Gaussianity, nonlinearity, and high dimensionality, which unfortunately, usually precludes analytic solutions. It is a strong belief that the issue of state measurement ultimately remains best handled within the framework of statistical inference. Instead of using

linearization techniques, the estimation problem is solved directly with Bayesian methodology.^{2,3} However, the Bayesian paradigm involves calculation of high-order integrals of the time state estimation. Thus, in the last few decades, many approximation filtering schemes, which are well known as the methods of particle filtering (PF), also known as condensation or sequential Monte Carlo methods (SMC),⁴⁻⁷ have been developed to seek a simulation-based way to surmount the problems.

However, a general 3D tracking problem with 6-DOF often requires thousands of particles,⁸ which can run foul of computational complexity and further interfere real-time performance for tracking agile motion. Moreover, degeneracy phenomenon is a common problem with particle filters. As a result of degeneracy, all but one particle will have negligible weight after a few state transitions. Degeneracy implies the wastage of computational resources that a large effort is engaged to update particles whose contribution to the approximation to posterior states is almost zero. Doucet has shown that the variance of the importance weights can only increase over time, so that degeneracy is an inevitable phenomenon with general sequential importance sampling scheme.⁹ There are commonly three methods to tackle the degeneracy problem¹⁰: (1) brute force approach, (2) good choice of importance density, and (3) use of resampling. The brute force approach uses a large enough sampling size to cover the effect of weight degeneration. However, it is often impractical in real-time estimation system. The method of choosing the optimal importance density can maximize the effective sampling size,⁹⁻¹² which is a suitable measure of degeneracy. The third method involves using the resampling process to reduce degenerate effects.⁴ Although resampling has been employed a lot in generic particle filter to avoid degeneracy as one of the most popular methods, it introduces additional computation complexity and cannot help to reduce the number of particles.

In this paper, we intent to explore possible ways to optimize the particle filter for enhancing 3D tracking performances. On the one hand, a new data fusion method is proposed to obtain the optimal importance density function, so that particle crowds can represent the posterior states in a much more efficient fashion. As a result, for achieving the same tracking accuracy, the number of particles used in 3D tracking is greatly reduced. On the other hand, we develop a method for reconfigurable vision systems to maximize the effective sampling size in particle filter, which consequentially helps to solve the degeneracy problem and minimize the tracking error.

2. Developing the Particle Framework

At time step k , when a measurement \mathbf{y}_k becomes available, according to the Bayes' rule,² the posterior probability function of the state vector can be calculated using the following equation

$$p(\mathbf{x}_k | \mathbf{y}_{1:k}) = \frac{p(\mathbf{x}_k | \mathbf{x}_{k-1})p(\mathbf{y}_k | \mathbf{x}_k)}{p(\mathbf{y}_k | \mathbf{y}_{1:k-1})}, \quad (1)$$

$P(\mathbf{y})$ is a normalizing constant and Eq. (1) can be written as

$$p(\mathbf{x}_k | \mathbf{y}_{1:k}) \propto p(\mathbf{x}_k | \mathbf{x}_{k-1}) p(\mathbf{y}_k | \mathbf{x}_k). \quad (2)$$

Suppose that at time step k , there is a set of particles, $\{\mathbf{x}_k^i, i = 1, \dots, N_s\}$, with associated weights $\{w_k^i, i = 1, \dots, N_s\}$ randomly drawn from importance sampling,^{9,11} where N_s is the total number of particles. The weight of particle i can be defined as

$$w_k^i \propto w_{k-1}^i \frac{p(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i) p(\mathbf{y}_k | \mathbf{x}_k^i)}{q(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i, \mathbf{y}_{1:k})}, \quad (3)$$

where $q(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i, \mathbf{y}_{1:k})$ is the importance density function. In this paper, we use the transition prior $p(\mathbf{x}_k | \mathbf{x}_{k-1})$ as the importance density function. Then, Eq. (3) can be simplified as

$$w_k^i \propto w_{k-1}^i p(\mathbf{y}_k | \mathbf{x}_k^i). \quad (4)$$

Furthermore, if we use Grenander's factored sampling algorithm,¹² Eq. (4) can be modified as

$$w_k^i = p(\mathbf{y}_k | \mathbf{x}_k^i). \quad (5)$$

The particle weights then can be normalized using

$$w_k^{*i} = \frac{w_k^i}{\sum_{i=1}^{N_s} w_k^i} \quad (6)$$

to give a weighted approximation of the posterior density in the following form

$$p(\mathbf{x}_k | \mathbf{y}_{1:k}) \approx \sum_{i=1}^{N_s} w_k^{*i} \delta(\mathbf{x}_k - \mathbf{x}_k^i), \quad (7)$$

where δ is the Dirac's delta function.

3. Data Fusion for Importance Density Optimization

3.1. Methodology

3.1.1. Data fusion algorithm and the pseudo likelihood

The 3D tracking task here is performed with an *active vision system*¹³ using pattern projection, which is similar to a passive stereo vision system with one of the cameras replaced by a projector. Using a color-encoded structured light pattern,¹⁴ the active vision system can yield good results in 3D visual sensing with a single view.

In order to obtain better expression of posterior states, the importance density function should be moved toward the region of high likelihood. Notwithstanding, because the support variables are different, likelihood functions cannot be used directly to modify the importance density. To surmount this problem, a pseudo

likelihood function is first generated with the latest passive sensing data. Then, the pseudo likelihood is projected to the importance density space and the importance density is modified by fusing the sensing data in it. The pseudo likelihood function is generated with the most current observation of certain reference feature points through passive sensing. It is a subset of the likelihood function and it can represent the likelihood function to certain extent. The advantage of the pseudo likelihood function is that it can be projected to the importance density space easily by using the inverse procedure of passive sensing observation model (a monocular camera model). The basic idea of the proposed approach is illustrated in Fig. 1. Suppose that in a set of reference points, the observation of passive sensing (of a monocular camera) can be expressed as a function of the current state with noise

$$y_{k_j}^R = g_j(x_k, \zeta), \quad j = 1, \dots, N_R, \tag{8}$$

where ζ is noise and N_R is the number of reference points.

Equation (8) can be looked on as a pseudo likelihood function

$$y_k^R \sim \hat{L}_y = P(\mathbf{y}|x_k^R). \tag{9}$$

On the contrary, the current state can be estimated using the inverse function of Eq. (8) as

$$x_k = g^{-1}(x_{k-1}, y_k^R, \zeta), \tag{10}$$

which is in fact a projection of the pseudo likelihood to the importance density (x_k) space

$$\hat{L}_x = P(x_k|x_{k-1}, y_k^R, \zeta). \tag{11}$$

Then, Eq. (11) can be used to achieve the algorithm for data fusion as shown in Table 1.

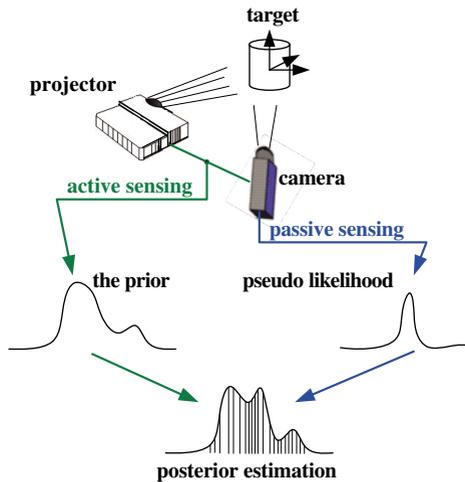


Fig. 1. Data fusion with updated passive sensing data.

Table 1. Passive data fusion algorithm in particle filter.

Assume that at the previous state $k - 1$, we have the particle crowd $\{x_{k-1}^i, w_{k-1}^i\}_{i=1}^N$, then proceed as following at time k

1. Sampling: simulate $x_k^i \sim P(x_k|x_{k-1})$.
2. Calculate weights: compute the weights according to likelihood function and conduct normalization.
3. Pseudo likelihood computation: calculate $\hat{L}_x = P(x_k|x_{k-1}, y_k^R, \zeta)$.
4. Data fusion: simulate x_k^i , draw αN samples from the prior $P(x_k|x_{k-1})$ and $(1 - \alpha)N$ samples from the pseudo likelihood projection $P(x_k|x_{k-1}, y_k^R, \zeta)$, where α is a data fusion factor, $0 \leq \alpha \leq 1$.
5. Update weights: compute the weights according to the new likelihood function and conduct normalization.
6. Resampling.

3.1.2. Importance density optimization

Degeneracy is a common phenomenon with particle filters.⁹ As a result of degeneracy, all but one particle will have negligible weight after a few state transitions. Degeneracy implies the wastage of computational resources that much effort is devoted to updating particles whose contribution to the approximation to posterior states is almost zero.

We can here adopt the *effective sampling size* N_k^{eff} ,¹⁵ which is a suitable measure of degeneracy, as a criterion to guide the optimization process for data fusion. As N_k^{eff} cannot be evaluated exactly,¹⁰ an estimate \hat{N}_k^{eff} of N_k^{eff} can be calculated by

$$\hat{N}_k^{eff} = \frac{1}{\sum_{i=1}^N (\tilde{w}_k^i)^2}, \quad (12)$$

where \tilde{w}_k^i is the normalized weight indicated in Eq. (6).

A large N_k^{eff} implies that the likelihood is located closely to the prior $P(x_k|x_{k-1})$, so that the particle crowd from the prior x_k^i can be relied on better. Thus, the percentage of effective sampling, \hat{N}_k^{eff}/N , can be used to defined to data fusion factor α (see Table 1) for importance density optimization by choosing

$$\alpha = \rho \frac{\hat{N}_k^{eff}}{N}, \quad (13)$$

where ρ is a positive scale factor, $\rho \geq 1$.

According to our previous study, when the configuration of the vision system (i.e., relative location to the object, optical and physical parameters of the camera and the projector) is not well designed, the percentage of effective sampling, \hat{N}_k^{eff}/N , can be very small, even may drop down to 5% sometimes. Since the effective sampling size is corresponding to the tracking error to some extent,¹⁶ a small effective sampling size may cause a large tracking error by particle estimation. In this case, the transition prior $P(x_k|x_{k-1})$ is not suitable to guide the importance density on

its own. The pseudo likelihood, which represents the most current sensing data, will help the PF obtain better sampling and reduce tracking error.

The value of ρ can be determined empirically, or it can simply be chosen as 1 (one). Then, the new particle crowd after data fusion is

$$x_k^{*i} = \alpha x_k^i + (1 - \alpha) \hat{x}_k^i, \tag{14}$$

where x_k^i is drawn from the prior and \hat{x}_k^i is drawn from the pseudo likelihood projection.

3.2. Simulation results

Figure 2 shows the estimation error for 3D location tracking by different methods. The generic PF, which employed 800 particles, performed the best, while the extended Kalman filter (EKF), with 100 particles, performed the worst because of its disadvantage in dealing with multi-modality. The proposed PF with data fusion, even only with 100 particles, achieved performance approximately as good as the generic PF.

With a well-expressed importance density, the proposed PF can achieve better real-time performance with expedition. Simulation results demonstrate the superiority of the proposed method in comparisons with EKF and generic PF (GPF). Because the EKF does not involve calculations of sampling, it can achieve faster real-time performance with an average 0.0146s for each state. The algorithm running times for GPF with 200, 400, and 800 particles are 0.0271, 0.0522, and 0.0998s

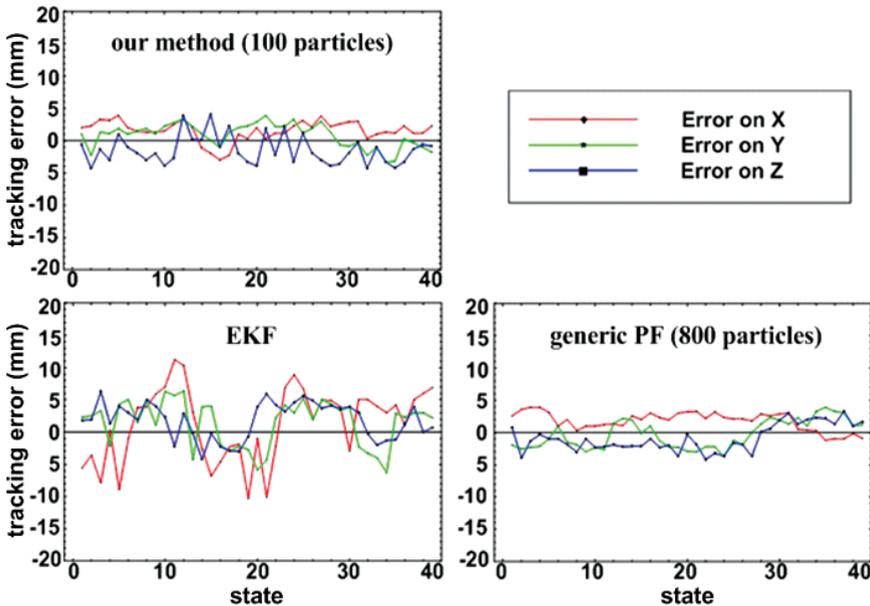


Fig. 2. Tracking accuracy comparison.

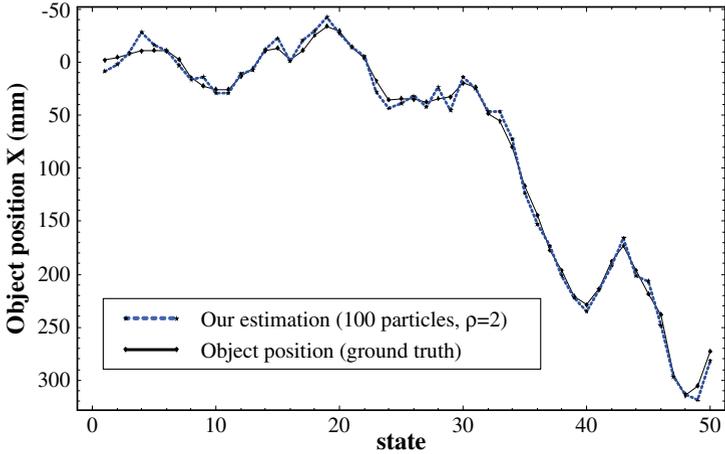


Fig. 3. Visual tracking using a PF with 100 particles and $\rho = 2$.

respectively. With only 100 particles, the proposed PF excels the GPFs in running time and only spent 0.0150s in average.

We then compared the tracking errors with different data fusion factors. A PF with 100 particles was adopted for the tracking shown in Fig. 3. In this simulation, the average percentage of effective sampling, \hat{N}_k^{eff}/N (before resampling), is about 36%.

As shown in Fig. 4, the data fusion factor affects tracking performance. According to our simulation study, even with the same vision system configuration, the effective sampling size may change during the tracking process and it is not

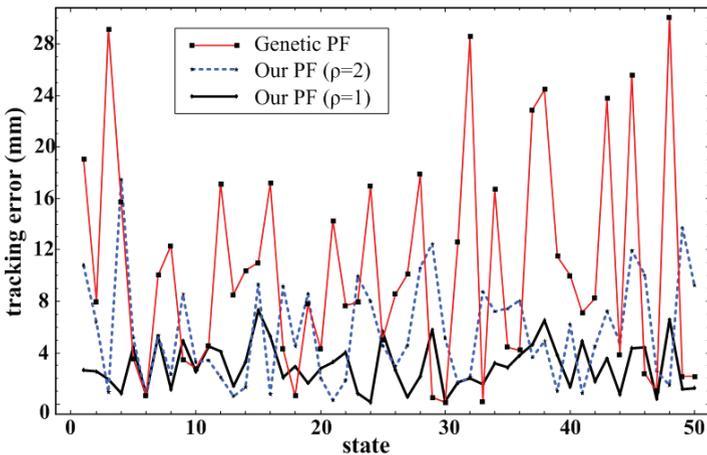


Fig. 4. Tracking error comparison.

Table 2. Some choices of ρ or α .

\hat{N}_k^{eff}/N value	< 0.05	0.1–0.2	0.2–0.3	> 0.3
ρ or α value	$\alpha \approx 0.1$	$\rho \approx 3$	$\rho \approx 2$	$\rho \approx 1$

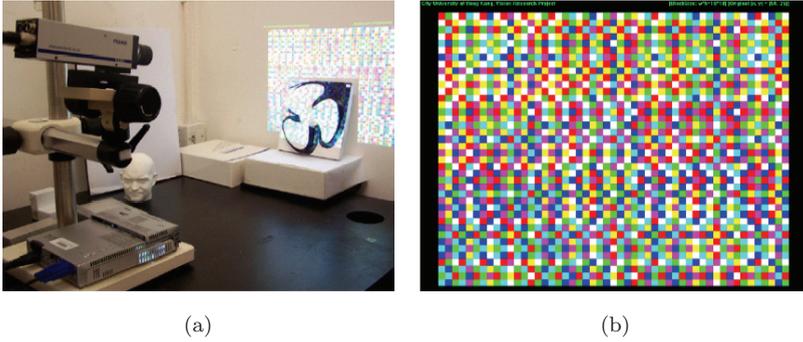


Fig. 5. The active vision system using color-encoded structured light.

necessary to adopt the same ρ when performing visual tracking. Table 2 indicates some possible choices of ρ (or α) according to our simulation results.

3.3. *Experimental results*

The proposed tracking method was tested with an active vision system that consists of a PULNIX TMC-9700 CCD camera and a PLUS V131 DLP projector (as shown in Fig. 5(a)). When the system is used in a visual tracking task, the projector projects a color-encoded structured light (see Fig. 5(b),¹⁷) onto the surface of the target object. Via triangulation, the system returns a time sequence of 3D object positions and orientations. This provides the measurement (given in Sec. 2) for the tracking formulation.

We used a concave object as the target (Fig. 6), which was moved arbitrarily by hand in 3D space to give motions with 3-DOF translational and 2-DOF rotational. The tracker (formulated in Secs. 2 and 3) was used to estimate the target's 5-DOF position. Here, a PF with 100 particles was employed. Since the object was moved randomly and the tracking was performed in real-time, quantitative results on tracking accuracy were hard to obtain due to the lack of the ground truth. We thus re-projected the estimated object positions and orientations onto the camera image (the red circles shown in Fig. 6) for qualitative evaluation. Some examples of snap shots in the tracking are shown in Fig. 6. With a sampling rate of about 12 fps, correct and reliable trackings were observed in the implementation. In Fig. 6, the tracking errors were mainly caused by the sensing itself, rather than the tracker. For example, in the frame shown in the bottom-right, relatively larger tracking error is observed. This is because the target happened to move to a position where the structured light pattern could not be detected clearly.

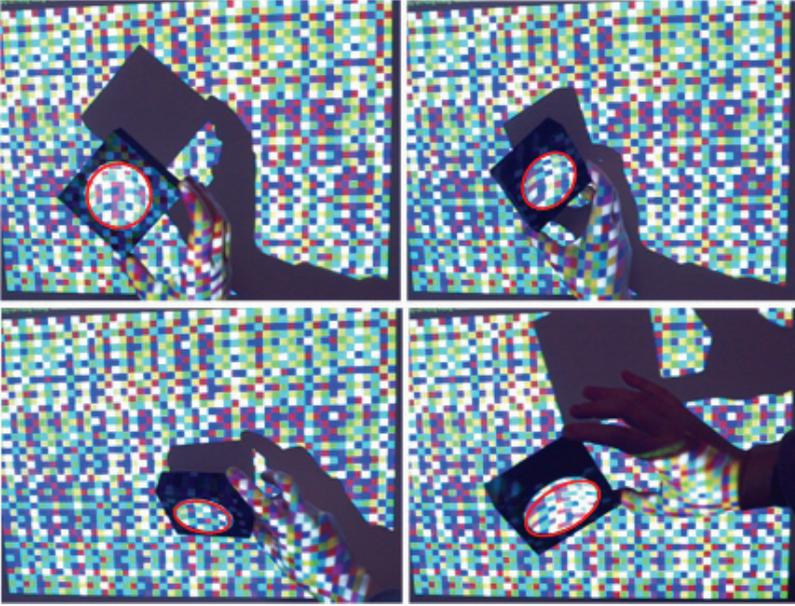


Fig. 6. Tracking a concave object with the proposed method.

4. Dynamic View Planning for Maximizing the Number of Effective Particles

4.1. Methodology

The 3D tracking task here is performed with a *reconfigurable vision system*.¹⁶ In this section, we intend to use the reconfigurability of the vision system to reduce those effects of degeneracy in particle filter. According to Refs. 11 and 12, the effective sampling size N_k^{eff} at state k is defined as

$$N_k^{eff} = \frac{N_s}{1 + \text{Var}(w_k^i)}, \quad (15)$$

where w_k^i is referred to as the “true weight” indicated in Eq. (4) and N_s is the number of samples. As N_k^{eff} cannot be evaluated exactly,¹⁰ an estimate \hat{N}_k^{eff} of N_k^{eff} can be calculated by

$$\hat{N}_k^{eff} = \frac{1}{\sum_{i=1}^{N_s} (w_k^{*i})^2}, \quad (16)$$

where w_k^{*i} is the normalized weight indicated in Eq. (15).

We then define the rate of effective particles as

$$\lambda_k^{eff} = \frac{\hat{N}_k^{eff}}{N_s}. \quad (17)$$

Table 3. Dynamic view planning algorithm.

Assume that at the previous state $k - 1$, we have the particle crowd $\{\mathbf{x}_{k-1}^i, w_{k-1}^{*i}\}_{i=1}^{N_s}$ and a viewpoint configuration ζ_{k-1}^* , then proceed as following at state k

DO when $\zeta_k \in C_v^k$ (search the k th viewpoint configuration space)

{

Sampling:

Simulate $\{\mathbf{x}_k^i, i = 1, \dots, N_s\}$

Weights calculating and normalizing:

Make observation at each ζ_k , calculate w_k^i and w_k^{*i} according to (5), (6), calculate the rate of effective particles λ_k^{eff}

}

View planning:

Use the least-squared method to choose the best view plan ζ_k^* according to (18), and mark its corresponding particles and particle weights

Estimation:

Use the marked particles and weights corresponding to ζ_k^* to calculate $p(\mathbf{x}_k | \mathbf{y}_{1:k})$ according to (7)

View plan executing:

Move the current viewpoint to ζ_k^*

Then continue to make observations and calculations at state $k + 1$.

Finally, the view planning¹⁶ task is achieved by computing the best configuration ζ_k^* in the viewpoint configuration space C_v^k of the camera through the following equation

$$\zeta_k^* = \arg_{\zeta_k} \min \left(\sum_{i=1}^{N_s} (w_k^{*i})^2 \right) \Bigg|_{\zeta_k \in C_v^k} = \arg \max (\hat{N}_k^{eff}) |_{\zeta_k \in C_v^k}. \quad (18)$$

Our dynamic view planning algorithm is described in Table 3. At state k , first, for every candidate of viewpoint configuration ζ_k , we sample \mathbf{x}_k^i from the prior $P(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i)$ and calculate its corresponding particle weights w_k^{*i} and rate of effective particles λ_k^{eff} . Then, we use the least-squared method to search for the best configuration ζ_k^* that maximizes λ_k^{eff} and mark its particle weights and samples. Finally, these samples and weights are used to calculate the posterior by a weighted sum indicated in Eq. (7).

4.2. Simulation results

In the first simulation, we compared the effective particles (sample size) \hat{N}_k^{eff} of the proposed view planning method to the generic particle filter with a fixed viewpoint. We used 100 particles for each method and ran the simulation for 100 times with nine state transitions. The average number of effective particles of the two methods is plotted as shown in Fig. 7. The generic PF without view planning obtained a very low rate of effective particles (7%), while our approach maximized the rate of effective particles $\overline{\lambda^{eff}}$ at about 53% via dynamic view planning. Our algorithm

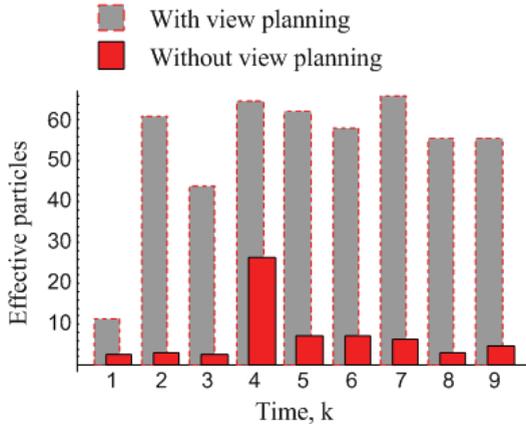


Fig. 7. Number of effective particles.

realizes view planning and achieves the best configurations of the vision system by maximizing \hat{N}_k^{eff} (or minimizing $\sum_{i=1}^{N_s} (w_k^{*i})^2$). In the second simulation, we checked tracking errors of different viewpoint configurations (camera locations) to prove that the best configuration in the sense of sampling efficiency is consistent with the best configuration in the sense of minimizing tracking error. We tested view planning results with different position parameters. As shown in Fig. 8, the average values of $\sum_{i=1}^{N_s} (w_k^{*i})^2$ and tracking errors are plotted with different configurations and estimation states. Here, the tracking error is defined as the distance between the estimated location and its true location. 100 tests each with 100 particles were employed. Nine viewpoint locations in camera coordinates, $x_i = 170 \times i$ (mm), when $i = 1, \dots, 9$, were employed. These locations were chosen empirically considering both the sensitivity and the kinematics constraints of the system.

It can be seen in the Fig. 8 that these two evaluation criteria shared the same tendency in viewpoint configuration. The comparison in 2D figure at the 6th estimation state is shown in Fig. 9. Different evaluation values with their values of standard deviation are plotted. Figure 9 clearly shows that tracking error reaches

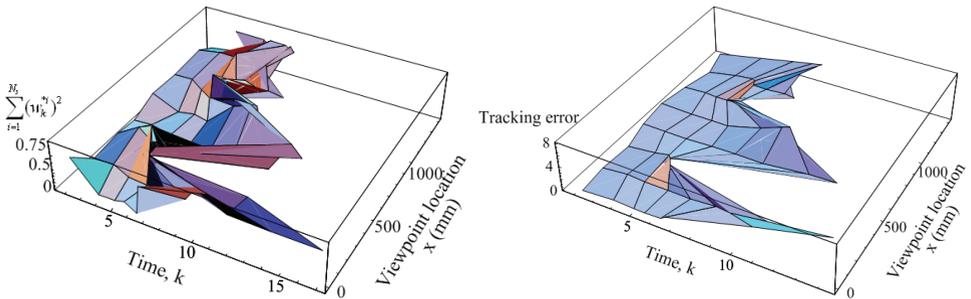


Fig. 8. Testing different evaluation criteria with position parameter $x(3D)$.

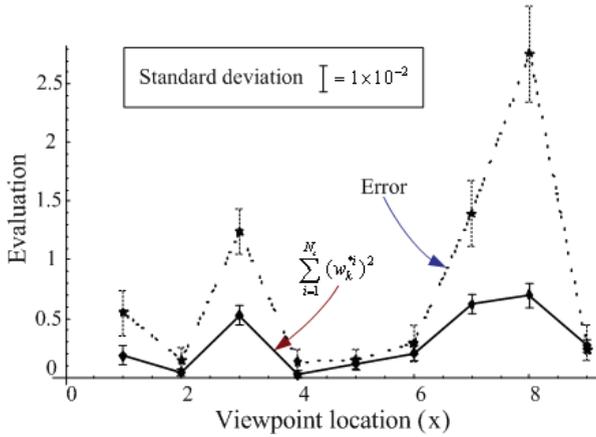


Fig. 9. Testing different evaluation criteria with position parameter x (2D).

its minimum (at the 4th x location, $x = 680$ mm) when $\sum_{i=1}^{N_s} (w_k^{*i})^2$ reaches its minimum value. In other words, the view planning driven by optimizing particle sampling actually minimizes the tracking error and improves tracking performance.

Then, we compared our method with the “centering” view planning method and the “error-orientated” view planning method. The centering method has been usually adopted in visual servoing,¹⁸ which controls the viewpoint to keep the image feature of the target object always at the center point of the image screen. In the “error-orientated” view planning method,¹⁹ the view planning process was directly driven by minimizing the estimated tracking error. The tracking errors of the aforementioned methods are plotted in Fig. 10, and their tracking performances are

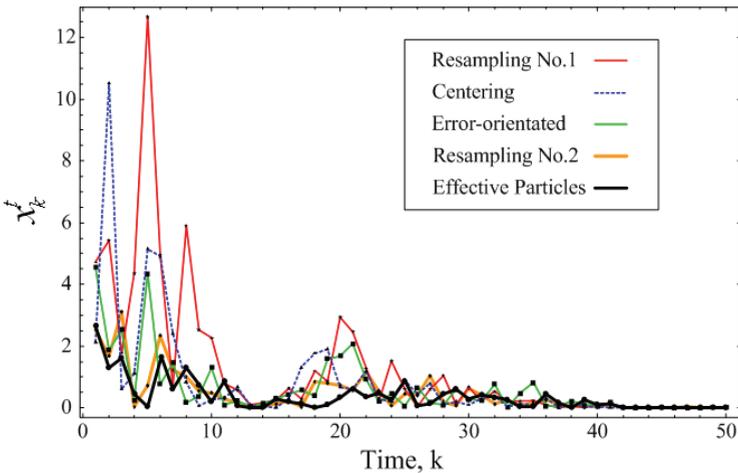


Fig. 10. Tracking errors using different methods.

Table 4. Tracking performance with different methods.

Method	Evaluation			
	Total number of particles	Average relative tracking error (%)	Average estimation time (s)	Average rate of effective particles (%)
Effective particles	100	2.7	0.032	49.5
Resampling PF No. 1	100	7.5	0.030	6.6*
Resampling PF No. 2	1000	4.5	0.266	7.6*
Centering	100	6.5	0.027	6.1
Error-orientated	100	5.5	0.111	7.1

(*This is the rate of effective particles λ^{eff} before resampling. After resampling, the rate of effective particles is compulsory modified to 100%. Even though it shows no improvement in tracking performance).

evaluated in Table 4. In these tests, our view planning method by effective particles was superior to others in tracking performance with the smallest tracking error and reasonable tracking speed. When the resampling method was used, although it could reduce the effects of weights degeneracy, the tracking error was large when using a small number of particles, whereas tracking speed was slow when using a large number of particles. Without testing the particle weights to obtain the best configuration, the centering method showed its advantage in tracking speed. However, because it can only compensate for a part of the sensing error and because it still suffers from particle degeneracy, its tracking error was larger than our method. The error-orientated method was supposed to be the one that could achieve good results in tracking accuracy. However, practically, it needs direct backward calculations and random search in the parameter space that may preclude unique solutions and yet, cannot overcome the degeneration, which affects its tracking accuracy and the achievable tracking speed as well.

Our method with effective particles minimizes tracking error by revealing the system to a better swarm of importance samples and interpreting the posterior state in a better way. Furthermore, it reduces particles' degeneracy significantly, so that a relative smaller particle crowd can be used to achieve the same level of tracking performance, and thus increases possible tracking speed.

4.3. Experimental results

The implementation of the proposed view planning method was conducted using our reconfigurable vision system, with a PC-based IM-PCI system and a variable scan frame grabber. This system supports many real-time processing functions including some feature extraction such as edge detection. Our algorithms were developed in VC++ programming language and run as imported functions by ITEX-CM. The system setup consists of a color CCD camera (model Pulnix TMC-6), with a resolution of 640×480 pixels, a pan-tilt unit (model PTU-46-17.5) for two-axis angular motion, and a linear motion system with a guideway (model KK86-20) and

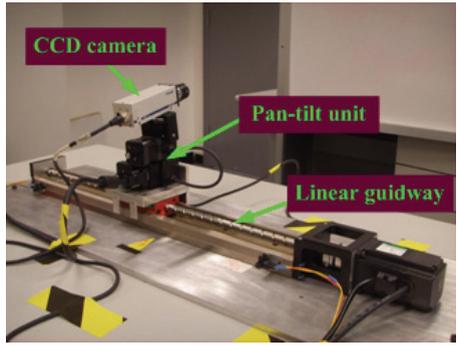


Fig. 11. Reconfigurable vision system with 3 DOFs.

motion controller (model Elmo BAS-3/320-2). A photo of this 3-DOF system is shown in Fig. 11.

For simplicity's sake, we used a point object for the experiment and made the object undergo uniform circular motion around the center with different diameters on a plane perpendicular to the optical axis of the camera at its original location (see Fig. 12). Using this motion, we can eliminate both influence of image feature location and influence of velocity on tracking error.¹

The average tracking errors on different concentric circles with different methods were calculated and are given in Table 5. These results show that our view planning method with effective particles can achieve small tracking error then other two methods.

Average tracking speeds with different methods are listed in Table 6. Because our method and centering method do not involve random search procedure that the error-orientated method uses, they both achieved very nice real-time performance.

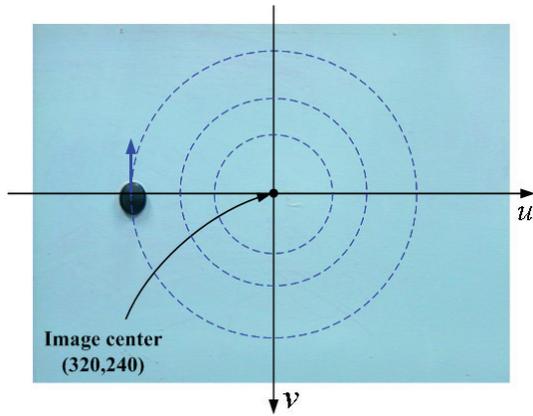


Fig. 12. Uniform circular motion at different diameters.

Table 5. Absolute tracking errors with different view planning methods.

Test diameter $2r$ (pixel)	Effective particles		Centering		Error-orientated	
	Mean tracking error (mm)	Standard deviation (mm^2)	Mean tracking error (mm)	Standard deviation (mm^2)	Mean tracking error (mm)	Standard deviation (mm^2)
400	6.2	19.1	7.2	18.5	8.8	21.6
360	6.1	21.1	9.1	19.7	9.0	19.1
320	4.1	18.8	8.3	16.1	8.9	19.6
280	5.6	17.8	6.6	18.8	7.0	19.6
240	5.9	19.5	6.5	15.5	5.5	18.5
200	5.3	19.3	6.2	16.5	8.4	19.5
160	4.6	17.1	7.0	17.3	4.7	18.1
120	5.2	16.6	6.3	15.6	5.9	19.6
80	5.8	18.3	7.5	16.1	5.9	18.6
Mean	5.4	18.6	7.2	17.1	7.1	19.4

Table 6. Tracking speed with different view planning methods.

Method	Effective particles	Centering	Error-orientated
Tracking speed (fps)	24	24	7

Our method with effective particles was then implemented to track a pen tip that was moving randomly with an average speed about 1 m/s. The pen tip was detected and tracking based on the segmentation with color and contour cues. Some examples of snapshots in the tracking with their corresponding viewpoint locations (best configurations) are shown in Fig. 13. In this experiment, every pair of two sequential frames was employed and compared to calculate the depth information z , and beside the current configuration information, the Chinese calligraphy background was used as correspondence between every two frames for further modification. Even with this process, a tracking rate of about 17 fps was achieved in the implementation. We then reprojected the estimated 3D locations of the pen tip onto the image space for tracking error analysis. The red target marks in Fig. 13

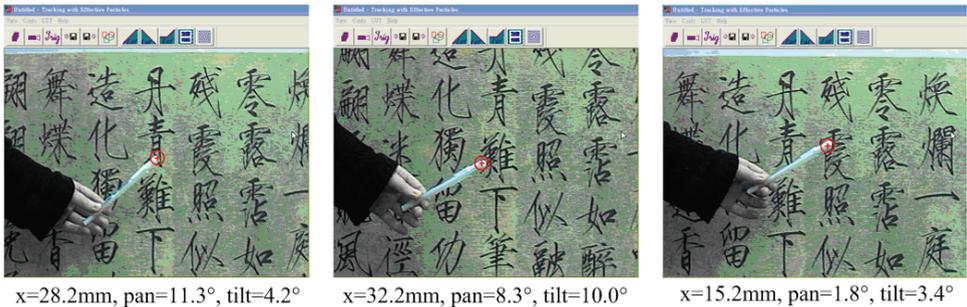


Fig. 13. Dynamic view planning in 3D tracking by our method.

represent those estimations from our tracking algorithm. Experimental results show that the tracking was conducted with very good accuracy, with an average tracking error of 2.8 pixels.

5. Conclusion

In this paper, two different methods for active vision and reconfigurable vision system are explored respectively to optimize particle filter for enhancing 3D tracking performance. A new data fusion method has been proposed to obtain the optimal importance density function for active vision, so that particle crowds can represent the posterior states in a much more efficient fashion. As a result, for achieving the same tracking accuracy, the number of particles used in 3D tracking is greatly reduced. We have also developed a method for reconfigurable vision systems to maximize the effective sampling size in particle filter, which consequentially helps to solve the degeneracy problem and minimize the tracking error. Simulation and experimental results have verified the effectiveness of our methods.

Acknowledgment

This work was supported by the Research Grants Council of Hong Kong (Project No. CityU 118311).

References

1. H. Y. Chen and Y. F. Li, Enhanced 3D tracking using nonsingular constraints, *Optical Engineering* **45**(10) (2006) 107202-1-12.
2. J. M. Bernardo and A. F. M. Smith, *Bayesian Theory* (Wiley, Chichester, 1994).
3. Y. Zhang and Q. Ji, Active and dynamic information fusion for multisensor systems with dynamic Bayesian networks, *IEEE Transactions on Systems, Man and Cybernetics (Part B)* **36**(2) (2006) 467-472.
4. N. J. Gordon, D. J. Salmond and A. F. M. Smith, Novel approach to nonlinear/non-Gaussian Bayesian state estimation, *IEEE Proceedings for Radar and Signal Processing* **140**(2) (1993) 107-113.
5. A. Doucet, N. D. Freitas and N. Gordon, *Sequential Monte Carlo Methods in Practice* (Springer-Verlag, New York, 2001).
6. M. Isard and A. Blake, CONDENSATION — conditional density propagation for visual tracking, *International Journal of Computer Vision (IJCV)* **29**(1) (1998) 2-28.
7. W. R. Scott *et al.*, Performance-orientated view planning for automatic model acquisition, in *Proceedings of the 31th International Symposium on Robotics* (2000), pp. 314-319.
8. D. Fox, KLD-sampling: Adaptive particle filters, *Proceedings of Conference on Neural Information Processing Systems* (2001).
9. A. Doucet, On sequential simulation-based methods for Bayesian filtering, *Technical Report CUED/F-INFENG/TR 310* (Department of Engineering, Cambridge University, 1998).
10. M. S. Arulampalam *et al.*, A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking, *IEEE Transactions on Signal Processing* **50**(2) (2002) 174-188.

11. N. Bergman, *Recursive Bayesian Estimation: Navigation and Tracking Applications*, Ph.D. Dissertation (Linköping University, Linköping, Sweden, 1999).
12. J. Carpenter, P. Clifford and P. Fearnhead, Improved particle filter for nonlinear problems, *IEE Proceedings on Radar, Sonar, Navigation* **146**(1) (1999) 2–7.
13. P. Besl, Active optical range imaging sensors, *Machine Vision and Applications* **1** (1988) 127–152.
14. P. M. Griffin, L. S. Narasimhan and S. R. Yee, Generation of uniquely encoded light patterns for range data acquisition, *Pattern Recognition* **25**(6) (1992) 609–616.
15. J. S. Liu and R. Chen, Sequential Monte Carlo methods for dynamical systems, *Journal of American Statistical Association* **93** (1998) 1032–1044.
16. H. Y. Chen and Y. F. Li, Dynamic view planning by effective particles for 3D tracking, *IEEE Transactions on Systems, Man and Cybernetics (Part B)* **39**(1) (2009) 242–253.
17. B. Zhang, Y. F. Li and Y. Wu, Self-recalibration of a structured light system via plane-based homography, *Pattern Recognition* **40**(4) (2007) 1368–1377.
18. E. Malis, F. Chaumette and S. Boudet, 2-1/2-D visual servoing, *IEEE Transactions on Robotics and Automation* **15**(2) (1999) 238–250.
19. J. R. Spletzer and C. J. Taylor, Dynamic sensor planning and control for optimally tracking targets, *International Journal of Robotics Research* **22**(1) (2003) 7–20.



Huiying Chen received her bachelor's degree in mechatronics engineering from South China University of Technology, China, and her master's degree in precision machinery engineering from the University of Tokyo, Japan. She obtained the Ph.D. degree from the Department of Manufacturing Engineering and Engineering Management at City University of Hong Kong, Hong Kong. She is now with the Department of Industrial and Systems Engineering of the Hong Kong Polytechnic University. Her research interests include robot vision, visual tracking, and dynamic view planning.



Youfu Li received his B.S. and M.S. degrees in electrical engineering from Harbin Institute of Technology, China. He obtained his Ph.D. degree from the Robotics Research Group, Department of Engineering Sciences of the University of Oxford in 1993. From 1993 to 1995, he was a postdoctoral research staff in the Department of Computer Sciences, University of Wales, Aberystwyth, UK. He is currently an associate professor in the Department of Manufacturing Engineering and Engineering Management at City University of Hong Kong. His research interests include robot sensing, robot vision, sensor-based control, sensor-guided manipulation, 3D vision, visual tracking, mechatronics, and automation. He has served as associate editor of *IEEE Transactions on Automation Science and Engineering* and is associate editor for *IEEE Robotics and Automation Magazine*.