Stochastic Filtering for Motion Trajectory in Image Sequences Using a Monte Carlo Filter with Estimation of Hyper-Parameters

Naoyuki Ichimura

Information Technology Research Institute
National Institute of Advanced Industrial Science and Technology (AIST)
1-1-1, Umezono, Tsukuba, Ibaraki, 305-8568 Japan
nic@ni.aist.go.jp

Abstract

False matching due to errors in feature extraction and changes in illumination between frames may occur in feature tracking in image sequences. False matching leads to outliers in feature motion trajectory. One way of reducing the effect of outliers is stochastic filtering using a state space model for motion trajectory. Hyper-parameters in the state space model, e.g., variances of noise distributions, must be determined appropriately to control tracking motion and outlier rejection properly. Likelihood can be used to estimate hyper-parameters, but it is difficult to apply online tracking due to computational cost. To estimate hyper-parameters online, we include hyper-parameters in state vector and estimate feature coordinates and hyper-parameters simultaneously. A Monte Carlo filter is used in state estimation, because adding hyper-parameters to state vector makes state space model nonlinear. Experimental results using synthetic data show that the proposed method can estimate appropriate hyper-parameters for tracking motion and reducing the effect of outliers.

1 Introduction

Feature tracking is essential to image sequence analysis in computer vision. False matching due to errors in feature extraction and changes in illumination between frames may occur in feature tracking. False matching leads to outliers in the feature motion trajectory. The effect of outliers must be reduced to yield good data for subsequent processing such as shape from motion[1][2] and motion segmentation[3].

One way of reducing this effect is stochastic filtering using a state space model for motion trajectory, as typified by the Kalman filter[4]–[6]. Since state estimation result based on stochastic filtering varies with hyper-parameters, e.g., variances of noise distributions, in state space model, appropriate hyper-parameters should be used to carry out state estimation properly. Likelihood $p(\boldsymbol{y}_k \mid Y_{k-1})$, $k = 1, 2, \ldots, t$ of time series of measurement $Y_t = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_t\}$ can be used to estimate hyper-parameters, but it is difficult to apply tracking due to computational cost; numerical search like coarse-to-fine method for grid of hyper-parameters is necessary. Stochastic property of motion generally varies with time and lag is not allowed in tracking, thus online estimation of hyper-parameters is quite important to feature tracking.

The Multiple-Model adaptive filter[7] can be used for online estimation of hyper-parameters. Multiple models with different hyper-parameters are prepared, and state estimation results of these models are combined based on likelihoods of models for current measurement to use appropriate hyper-parameters. Since all combinations of hyper-parameters should be given in this method, if appropriate hyper-parameters are not in the given scope, the filter can not yield good state estimation results. In addition, a lot of models may be necessary when many hyper-parameters are needed and/or quantization of them must be fine to obtain good accuracy; it reduces efficiency of algorithm.

In this paper, we include hyper-parameters in state vector and estimate feature coordinates and hyper-parameters simultaneously online. Using this simultaneous estimation, we can adjust hyper-parameters to feature motion automatically. Since adding hyper-parameters to state vector makes state space model nonlinear, we need state estimation method for nonlinear model. The sequential Monte Carlo (SMC) method used for nonlinear/non-Gaussian state space models has been proposed recently[8]–[11], so the Monte Carlo filter (MCF)[9], a type of the SMC, is applied to state estimation of our model. Online estimation of hyper-parameters by adding them to state vector was proposed in 1970's, but the extended Kalman filter which needs linear approximation of model and assumption of Gaussian distri-

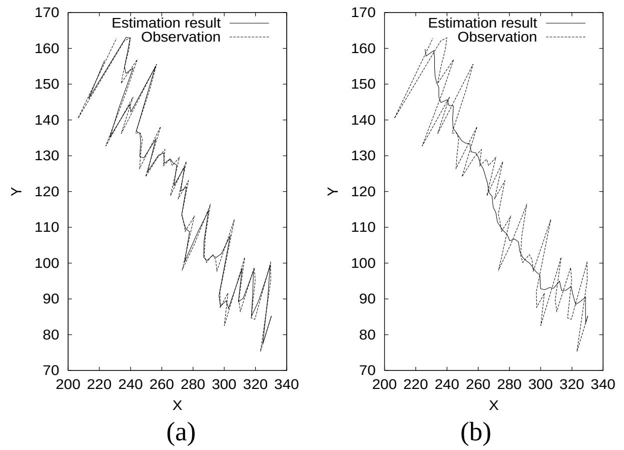


Figure 1. An example of the effect of hyper-parameters for state estimation. (a) $\tau^2 = 1.0$ and $\sigma^2 = 1.0$. (b) $\tau^2 = 0.01$ and $\sigma^2 = 50.0$.

bution for state and noises could not yield good state estimation results[12][13]. The approximation and assumption are not needed for the MCF.

The MCF allows us to use the non-Gaussian observation noise distribution which is useful to reduce the effect of outliers[14]. Using heavy-tailed non-Gaussian distribution, e.g., Cauchy distribution, we can represent both observation noise with high probability and outliers with low probability. Thus state estimation for motion trajectory with outliers can be done.

Adaptive filtering for motion trajectory with time-dependent statistical property can be realized using online estimation of hyper-parameters and non-Gaussian noise distribution mentioned above. Experiments using synthetic data are shown to discuss the usefulness of the proposed method.

2 State Space Model for Motion Trajectory

2.1 Notation and prior models

A state space model is represented as follows:

$$\boldsymbol{x}_t = \boldsymbol{F}\boldsymbol{x}_{t-1} + \boldsymbol{G}\boldsymbol{v}_t \tag{1}$$

$$\boldsymbol{y}_t = \boldsymbol{H}\boldsymbol{x}_t + \boldsymbol{w}_t \tag{2}$$

where Eq. (1) is the state transition equation, $\boldsymbol{x}_t$ is the state vector, and $\boldsymbol{v}_t$ is the system noise vector. Matrix $\boldsymbol{F}$ and $\boldsymbol{G}$ are system matrices. Equation (2) is the observation equation, $\boldsymbol{y}_t$ is the observation vector, and $\boldsymbol{w}_t$ is the observation noise vector. Matrix $\boldsymbol{H}$ is an observation matrix.

Stochastic filtering based on the state space model is used to estimate state vector $\boldsymbol{x}_t$ using time series of measurement $Y_t = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_t\}$, i.e., to calculate conditional probability density function $p(\boldsymbol{x}_t \mid Y_t)$.

The observation vector

$$\boldsymbol{y}_t = [x(t), y(t)]^T \tag{3}$$

represents coordinates of features in images. The following smoothness prior models, i.e., constant velocity model, are used for motion trajectory:

$$x(t+1) = 2x(t) - x(t-1) \tag{4}$$

$$y(t+1) = 2y(t) - y(t-1) \tag{5}$$

This assumption means that the change in velocity of feature between 2 adjacent frames is small and the velocity varies gradually in image sequence.

2.2 Noise distribution and hyper-parameters

Variables in system noise vector $\boldsymbol{v}_t$ are independent, and their distributions are represented as $q\left(v; m_q, \tau^2\right)$, where m_q and τ are parameters of location and scale. The multivariate form of system noise distribution is denoted as $q_v\left(\boldsymbol{v}_t; \boldsymbol{m}_{vq}, \boldsymbol{T}\right)$. Variables in observation noise vector $\boldsymbol{w}_t$ are independent, and their distributions are represented as $r\left(w; m_r, \sigma^2\right)$, where m_r and σ are parameters of location and scale. The multivariate form of observation noise distribution is denoted as $r_v\left(\boldsymbol{w}_t; \boldsymbol{m}_{vr}, \boldsymbol{\Sigma}\right)$. Since we assume the mean of noise is zero, the entities in $\boldsymbol{m}_{vq}$ and $\boldsymbol{m}_{vr}$ are zero.

Parameters τ^2 and σ^2 in system noise and observation noise distribution are called hyper-parameters, because their role is to control state estimation. Figure 1 shows an example of the effect of hyper-parameters for state estimation. The prior models of Eq.(4) and (5) and Gaussian distribution for system and observation noise were used. State estimation was done by the Kalman filter. The hyper-parameters, $\left(\tau^2, \sigma^2\right) = (1.0, 1.0)$ (Fig.1 (a)) and $(0.01, 50.0)$ (Fig.1 (b)), were used for the same observation. As we can see from Fig.1(a) and (b), the state estimation results are completely different. This example shows the importance of hyper-parameters in state estimation.

Likelihood $p(\boldsymbol{y}_k \mid Y_{k-1})$, $k = 1, 2, \ldots, t$, $Y_0 = \phi$ of time series of measurement can be used to determine hyper-parameters. It is difficult, however, to apply feature tracking due to computational cost. Since lag is not allowed in feature tracking and stochastic property of motion generally varies with time, online estimation of hyper-parameters is needed.

2.3 State space model to estimate hyper-parameters

From the prior models of Eq.(4) and (5), $x(t)$, $y(t)$, $x(t-1)$ and $y(t-1)$ are included in the state vector. And hyper-parameters are also included in the state vector to estimate them online as follows:

$$\begin{aligned}\boldsymbol{x}_t &= [x_s(t), y_s(t), x_s(t-1), y_s(t-1), \\ &\quad \log\tau^2(t), \log\sigma^2(t)]^T \end{aligned} \tag{6}$$

The logarithm of hyper-parameters is used to preserve scale parameters as positive. Since hyper-parameters are in the state vector, they are estimated online; both feature coordinates and hyper-parameters are estimated simultaneously to adjust the hyper-parameters to feature motion automatically.

Since hyper-parameters are included in the state vector and time-dependent to follow feature motion, we define the system matrices, observation matrix, system noise vector, and observation noise vector of the state space model as follows:

$$\mathbf{F}=\begin{bmatrix} 2 & 0 & -1 & 0 & 0 & 0 \\ 0 & 2 & 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{G}=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{7}$$

$$\boldsymbol{v}_t = [v_x(t), v_y(t), v_{\tau^2}(t), v_{\sigma^2}(t)]^T \tag{8}$$

$$\boldsymbol{w}_t = [w_x(t), w_y(t)]^T \tag{9}$$

where v_{τ^2} and v_{σ^2} are variables for system noise of hyper-parameters. The location and scale in multivariate form of system noise distribution $q_v(\boldsymbol{v}_t; \boldsymbol{m}_{vq}, \boldsymbol{T})$ are $\boldsymbol{m}_{vq} = [0,0,0,0]^T$ and $\boldsymbol{T} = \mathrm{diag}(\tau^2, \tau^2, \nu^2, \xi^2)$. The location and scale in multivariate form of observation noise distribution are $\boldsymbol{m}_{vr} = [0,0]^T$ and $\boldsymbol{\Sigma} = \mathrm{diag}(\sigma^2, \sigma^2)$.

ν^2 and ξ^2 are called hyper-hyper-parameters, because these parameters govern change in hyper-parameters.

3 State Estimation by Monte Carlo Filter

3.1 Monte Carlo filter algorithm

Adding hyper-parameters to state vector makes the state space model nonlinear, since system and observation noise vectors are the function of hyper-parameters in the state vector. We need state estimation method for such a nonlinear model, because the Kalman filter commonly used assumes a linear/Gaussian model.

The sequential Monte Carlo method (SMC) was proposed for nonlinear/non-Gaussian models[8]-[11]. The Monte Carlo filter (MCF)[9], a type of the SMC, is applied to state estimation of the proposed model. The procedure of the MCF is briefly described below.

In the MCF, probability distributions used in state estimation are approximated by m particles, i.e., m realizations from distributions, as follows:

$$\text{prediction}: p(\boldsymbol{x}_t \mid Y_{t-1}), \quad \left\{\boldsymbol{p}_t^{(1)}, \ldots, \boldsymbol{p}_t^{(m)}\right\}$$
$$\text{filter}: p(\boldsymbol{x}_t \mid Y_t), \quad \left\{\boldsymbol{f}_t^{(1)}, \ldots, \boldsymbol{f}_t^{(m)}\right\}$$

System noise distribution is approximated as follows:

$$\text{system noise}: q_v(\boldsymbol{v}_t; \boldsymbol{m}_{vq}, \boldsymbol{T}), \quad \left\{\boldsymbol{v}_t^{(1)}, \ldots, \boldsymbol{v}_t^{(m)}\right\}$$

State estimation is carried out by repeating the following one-step-ahead prediction and filter process:

[MCF algorithm]

[Step 1: generation of particles of initial distribution] Generate m particles, i.e., m random vectors, from an initial distribution of state $p_0(\boldsymbol{x})$. These particles $\boldsymbol{f}_0^{(i)}$ $(i = 1, \ldots, m)$ are regarded as initial filter distribution.

[Step 2: filtering] Repeat next steps.

[Step 2-1: generation of particles for system noise] Generate m particles $\boldsymbol{v}_t^{(i)} \sim q_v(\boldsymbol{v}_t; \boldsymbol{m}_{vq}, \boldsymbol{T})$, $i = 1, \ldots, m$.

[Step 2-2: one-step-ahead prediction] Compute particles, $\boldsymbol{p}_t^{(i)}$, representing the prediction distribution $p(\boldsymbol{x}_t \mid Y_{t-1})$ using the state transition equation:

$$\boldsymbol{p}_t^{(i)} = \boldsymbol{F}\boldsymbol{f}_{t-1}^{(i)} + \boldsymbol{G}\boldsymbol{v}_t^{(i)}, \; i = 1, \ldots, m \tag{10}$$

[Step 2-3: calculation of likelihood of $\boldsymbol{p}_t^{(i)}$] Calculate the likelihood $\alpha_t^{(i)}$ of particle $\boldsymbol{p}_t^{(i)}$ using measurement vector $\boldsymbol{y}_t$ and observation noise distribution as follows:

$$\alpha_t^{(i)} = r_v\left(\boldsymbol{y}_t - \boldsymbol{H}\boldsymbol{p}_t^{(i)}; \boldsymbol{m}_{vr}, \boldsymbol{\Sigma}\right), \; i = 1, \ldots, m \tag{11}$$

[Step 2-4: calculation of filter distribution] Calculate particles $\boldsymbol{f}_t^{(i)}$ of filter distribution by resampling particles $\boldsymbol{p}_t^{(i)}$ in accordance with the following probabilities:

$$Pr\left(\boldsymbol{f}_t^{(i)} = \boldsymbol{p}_t^{(i)}\right) = \frac{\alpha_t^{(i)}}{\alpha_t^{(1)} + \ldots + \alpha_t^{(m)}}, \; i = 1, \ldots, m \tag{12}$$

3.2 Intuition for adaptation of hyper-parameters by the MCF

The likelihoods of particles used in resampling in Step 2-4 are determined by the difference between measurement and predicted position, and observation noise distribution, i.e., $r_v\left(\boldsymbol{y}_t - \boldsymbol{H}\boldsymbol{p}_t^{(i)}; \boldsymbol{m}_{vr}, \boldsymbol{\Sigma}\right)$ in Eq.(11). The particle with high likelihood survives the resampling, thus hyper-parameters in such particle (state vector) are chosen as good one for adapting filter to feature motion.

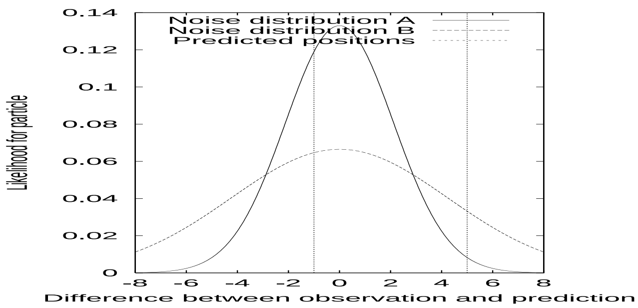


Figure 2. An example of observation noise distributions with different hyper-parameters.

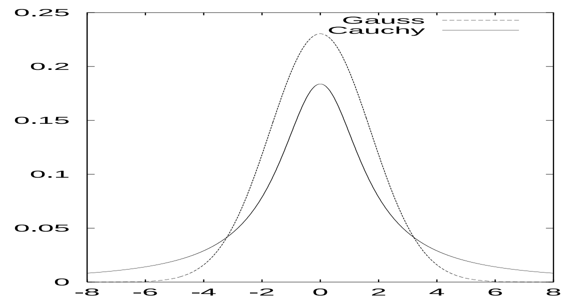


Figure 3. Gaussian distribution and Cauchy distribution; location: 0, scale: 3.

Figure 2 shows an example of two observation noise distributions with different hyper-parameters. If the difference between measurement and predicted position is -1, observation noise distribution A gives higher likelihood than one given by observation noise distribution B. So the hyper-parameters of distribution A is chosen with high probability. If the difference between measurement and predicted position is 5, the hyper-parameters of distribution B survives with high probability. Hyper-parameters are adapted to feature motion through such a selection mechanism in the MCF.

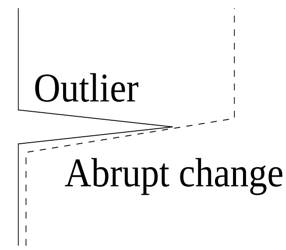


Figure 4. Tradeoff between outlier rejection and tracking abrupt changes in motion.

4 Non-Gaussian Observation Noise Distribution for Outlier Rejection

The MCF allows us to use the heavy-tailed non-Gaussian observation noise distribution which is useful to reduce the effect of outliers[14]. The following Cauchy distribution $C\left(0, \sigma^2\right)$ is selected for observation noise:

$$r\left(w; 0, \sigma^2\right) = \frac{\sigma}{\pi\left\{w^2 + \sigma^2\right\}} \tag{13}$$

where σ is scale parameter. It is heavy-tailed (Fig.3) and represents both observation noise with high probability and outliers with low probability. State estimation for motion trajectory with outliers, therefore, can be realized.

5 Experimental Results

The problem for controlling the tradeoff between tracking abrupt motion changes and outlier rejection is used to show the adaptation ability of the proposed filter.

Using heavy-tailed non-Gaussian distribution, we can realize state estimation for motion trajectory with outliers as mentioned in Section 4. A problem remains, however: the delay in tracking abrupt changes in feature motion (Fig.4). The purpose of this experiment is to control the tradeoff between outlier rejection and tracking abrupt motion changes. Tradeoff is controlled by changing hyper-parameters, i.e., scale parameters of observation noise distribution and system noise distribution. The larger scale parameter of system noise distribution, for example, enables higher tracking speed. Thus controlling tradeoff is equivalent to adjusting hyper-parameters based on feature motion.

Synthetic data shown in Fig.5 was used where the solid line represents the true trajectory and the dotted line represents observation with noise and outliers. This data contains 3 outliers ($t = 15, 30$ and 75) and an abrupt change in motion ($t = 50$). The Cauchy distribution $C\left(0, \tau^2\right)$ was also used for system noise to represent both smooth motion that may occur frequently and abrupt changes in motion that may occur infrequently.

To evaluate the best performance of the proposed filter, log-likelihood was used to determine hyper-hyper-parameters ν^2 and ξ^2[15]. Log-likelihood for the proposed filter can be computed using approximation as follows[9]:

$$l\left(\nu^2, \xi^2\right) \cong \sum_{t=1}^{N} \log\left(\sum_{i=1}^{m} \alpha_t^{(i)}\right) - N \log m \tag{14}$$

where $\alpha_t^{(i)}$ is likelihood calculated in Step 2-3 of the MCF algorithm and N is the length of the sequence of measurement Y_t. A coarse-to-fine method was used to determine hyper-hyper-parameters; first, coarse grid $\{1, 2, \ldots, 20\} \times \{1, 2, \ldots, 20\}$ was used as the candidate of parameters and

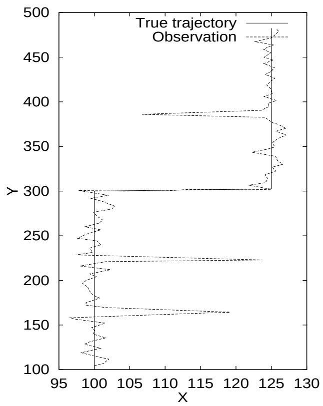


Figure 5. Synthetic data used in the experiment.

then a more detailed grid was selected based on the maximum log-likelihood in the coarse grid. Parameters determined for synthetic data are shown in Table 1.

The number of particles in the MCF, m, was 10,000. The initial distribution for coordinates of features was Gaussian with mean vector $\left[x\left(1\right), y\left(1\right), x\left(1\right), y\left(1\right)\right]^T$ and covariance matrix diag $(10, 10, 10, 10)$. The initial distribution for hyper-parameters was uniform distribution in $[-8, 8]$. Estimated coordinates of features were obtained from the mode of 2D distribution of $x_s\left(t\right)$ and $y_s\left(t\right)$ computed from particles $\boldsymbol{f}_t^{(i)}$ using a Parzen estimator[16]. The estimate of each hyper-parameter is also obtained using 1D distribution.

The proposed filter was compared to the Kalman filter for linear/Gaussian model which was obtained by removing hyper-parameters from state vector and using Gaussian distribution for both system and observation noise. Log-likelihood was also used to determine hyper-parameters τ^2 and σ^2 for the linear/Gaussian model to compare the best performance. Log-likelihood for the model is represented as follows:

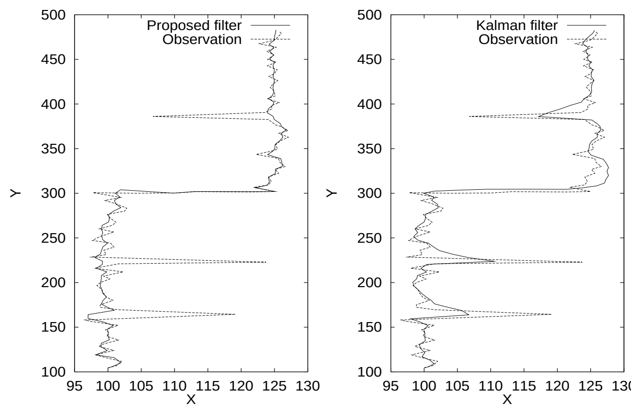
$$l_{LGM}\left(\tau^2, \sigma^2\right) = -\frac{1}{2}\left\{nN\log 2\pi + \sum_{t=1}^{N}\log\left|\boldsymbol{V}_{t|t-1}\right| + \sum_{t=1}^{N}\left(\boldsymbol{y}_t - \boldsymbol{m}_{t|t-1}\right)^T \boldsymbol{V}_{t|t-1}^{-1}\left(\boldsymbol{y}_t - \boldsymbol{m}_{t|t-1}\right)\right\} \quad (15)$$

where n is the number of dimensions of the state vector and $\boldsymbol{m}_{t|t-1}$ and $\boldsymbol{V}_{t|t-1}$ are the mean vector and covariance matrix of prediction distribution calculated in the Kalman filter. Determined parameters are shown in Table 1.

Figure 6 and 7 show the estimated trajectories and errors of coordinates. The effect of outliers was clearly reduced by the proposed filter (Fig.6 (a) and Fig.7 (a),(b)), while it was appeared in the results of the Kalman filter (Fig.6 (b) and Fig.7 (c),(d)). The tradeoff between tracking abrupt

Table 1. Hyper-hyper-parameters determined for the proposed filter, hyper-parameters determined for Kalman filer, and mean squared error between estimated and true trajectories of both filters.

Proposed filter			Kalman filer		
ν^2	ξ^2	MSE	τ^2	σ^2	MSE
0.006	0.034	0.118	0.20	8.5	0.269



(a) (b)

Figure 6. Filtering results for synthetic data. (a) proposed filter. (b) Kalman filter.

changes and outlier rejection was controlled properly by the proposed filter, because the scale parameter of system noise distribution, τ^2, was increased very rapidly at $t = 50$ (Fig.8). Since the larger scale parameter enables higher tracking speed, errors in coordinates of the proposed filter were smaller than that of the Kalman filter around $t = 50$ (Fig.7). Table 1 shows the mean squared error between estimated and true trajectories; the proposed filter was more accurate than the Kalman filter.

The other experimental results for shape from motion and an online implementation can be seen in the technical report which is a longer version of this manuscript [17]. It is available via the WWW: http://staff.aist.go.jp/naoyuki.ichimura/.

6 Conclusions

Stochastic filtering for a motion trajectory with estimation of hyper-parameters has been proposed. Hyper-parameters governing state estimation were included in state vector and estimated simultaneously with feature coordinates. Experimental result for synthetic data verified the

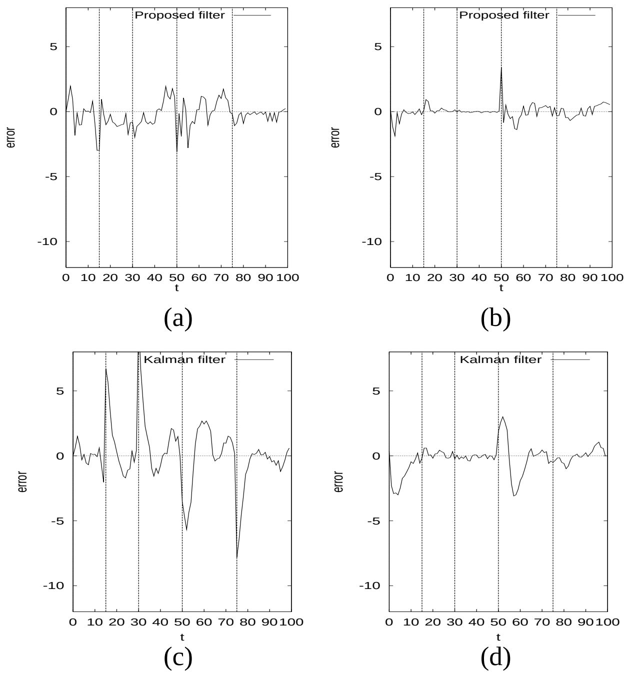


Figure 7. Error in coordinates for synthetic data. Abscissa represents time. (a),(b): error in x and y coordinates for the proposed filter. (c),(d): error in x and y coordinates for the Kalman filter. Vertical lines show when outliers (t=15,30,75) and abrupt change in motion (t=50) occur.

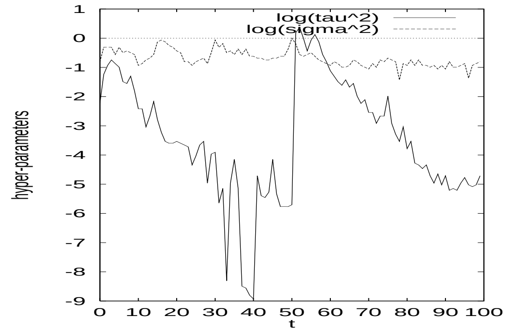


Figure 8. Changes in hyper-parameters, $\log 10\left(\tau^2\right)$ and $\log 10\left(\sigma^2\right)$, in the filtering.

usefulness of our proposed method.

One of the problems of the proposed filter is the effect of approximation of distribution by particles for state estimation. Estimation using particles sometime converges only on a number of varieties of particles; this phenomenon is called as "sample impoverishment"[18]. The algorithm for increasing varieties of particles should be incorporated to avoid the phenomenon and keep the adaptation ability of the filter especially for long image sequences. This is a projected work in our research.

References

[1] S. Christy and R. Horaud:"Euclidean reconstruction: from paraperspective to perspective," Proc. 4th European Conf. Comp. Vis., Vol.2, pp. 129-140, 1996.

[2] C. J. Poelman and T. Kanade: "A paraperspective factorization method for shape and motion recovery," IEEE Trans. Pattern Anal. & Mach. Intell., Vol.19, No.3, pp. 206-218, 1997

[3] N. Ichimura:"A robust and efficient motion segmentation based on orthogonal projection matrix of shape space," Proc. Int. Conf. on Computer Vision and Pattern Recognition, Vol.2, pp.446-452, 2000

[4] J.S. Meditch: "Stochastic optimal linear estimation and control," McGraw-Hill, 1969.

[5] E. D. Dickmanns and B. D. Mysliwetz: "Recursive 3-D road and relative ego-state recognition," IEEE Trans. Pattern Anal. & Mach. Intell., Vol.14, No.2, pp. 199-213, 1992.

[6] L. Goncalves, E. D. Bernardo, E. Ursella and P. Perona: "Monocular tracking of the human arm in 3D," Proc. 5th Int. Conf. Comp. Vis., pp. 764-770, 1995.

[7] Y. Bar-Shalom and T. E. Fortmann:"Tracking and data association," Academic Press, 1988.

[8] 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, No. 2, pp. 107-113, 1993.

[9] G.Kitagawa: "Monte Carlo filter and smoother for non-Gaussian nonlinear state space models," J. Comp. and Graph. Stat., Vol. 5, No. 1, pp. 1-25, 1996.

[10] M. Isard and A. Blake: "Condensation – conditional density propagation for visual tracking," Internat. J. Comp. Vis., Vol. 29, No. 1, pp. 5-28, 1998.

[11] J. S. Liu and R. Chen: "Sequential Monte Carlo methods for dynamic systems," J. Amer. Statist. Assoc., Vol. 93, pp. 1032-1044, 1998.

[12] B. D. O. Anderson and J. B. Moore:"Optimal Filtering," Prentice-Hall, 1979.

[13] G. Kitagawa: "A self-organizing state-space model," J. Amer. Statist. Assoc., Vol. 93, pp. 1203-1215, 1998.

[14] K. L. Lange, R. J. A. Little and J. M. G. Taylor:"Robust statistical modeling using the t distribution," J. Amer. Statist. Assoc., Vol.84, No.408, pp. 881-896, 1989.

[15] H. Akaike: "Likelihood and the Bayes procedures," Bayesian Stat., pp. 143-166, 1980.

[16] K. Fukunaga: "Introduction to Statistical Pattern Recognition," Academic Press, 1972.

[17] N. Ichimura:"Stochastic filtering for motion trajectory in image sequences using a Monte Carlo filter with estimation of hyper-parameters," tech. report AIST02-J-00002-3, Information Technology Research Institute, AIST, 2002

[18] O. King and D. A. Forsyth: "How does CONDENSATION behave with a finite number of samples?," Proc. 6th European Conf. Comp. Vis., pp. 695-709, 2000.