$\LaTeX$ Author Guidelines for CVPR Proceedings

Chen Du
PID:
@eng.ucsd.edu

Wenjun Zhang
PID: A53218995
wez177@eng.ucsd.edu

Jingyi Yang
PID:
@eng.ucsd.edu

Jinzhao Feng
PID:
@eng.ucsd.edu

Abstract

Abstract here.

1. Introduction

Introduction here.

2. Description of The Project

2.1. Kalman Filter

Kalman filter has been widely applied in object tracking. The Kalman filter is the optimal filter to estimate the state of a linear dynamic system when only noisy measurements are available[1]. Due to analytic tractability both operation and measurement noises are required to be Gaussian. In 1960, Ruodolf Kalman first introduced the system[2], and it was first applied in estimating the trajectory for the NASA Apollo program in 1969[3].

The operation of the filter basically consists of 2 steps:

1 Prediction, based on the last estimated state $\hat{x}_{k-1}$ and covariance matrix P_{k-1}:

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k \tag{1}$$

$$P_k^- = AP_{k-1}A^T + Q \tag{2}$$

where A is the time-transition matrix, u_k is the control or external input, B is a linear matrix which characterize the effects of the external input, and Q is the operation noise covariance matrix.

Equation (1) predicts the most probable state $\hat{x}_k^-$ based on the last estimation and external input. Equation (2) predicts the new covariance matrix P_k^- of the predicted state, given the previous estimated covariance matrix and operation noise.

2 Correction, based on the measurements to correct the prediction:

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \tag{3}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \tag{4}$$

$$P_k = (I - K_k H)P_k^- \tag{5}$$

where K_k is the Kalman gain, the term $z_k - H\hat{x}_k^-$ is called the innovation, H is the measurement matrix, and R is the measurement noise covariance matrix.

The Kalman gain K_k controls the effect that the current measurement has over the new state. Equation (4) and (5) correct the prediction based on the Kalman gain to estimate the state $\hat{x}_k$ and covariance matrix P_k.

Because the new estimations are generated based only on the previous state estimation and the current measurements, the Kalman filter is an online estimator. Besides, by increasing Q and R, P_k^- increases and K_k tends to zero, meaning that the last state becomes less important and the new measurement becomes less effective. It should also be noted that the transition matrix A and measurement matrix H in Kalman filter are certain, so when the model is nonlinear, no such certain matrices A and H exist, and Kalman filter no longer works.

2.2. Extended Kalman Filter

The Extended Kalman filter (EKF)[4][5] has been applied to alleviate the constraint of linearity. The basic framework of Extended Kalman filter is similar to that of Kalman filter, which is shown in Figure. 1. For convenience, the control input u_k has been set to zero. What?s different is that the Extended Kalman filter uses the linear function f and h to linearize the nonlinear transformations for state prediction and predicted measurement. And the transition matrix A and measurement matrix H are substituted with Jacobian matrices derived from the non-linear function f and h:

$$A = \frac{\partial f}{\partial x}|_{\hat{x}_{k-1}} \tag{6}$$

$$H = \frac{\partial h}{\partial x}|_{\hat{x}_k^-} \tag{7}$$

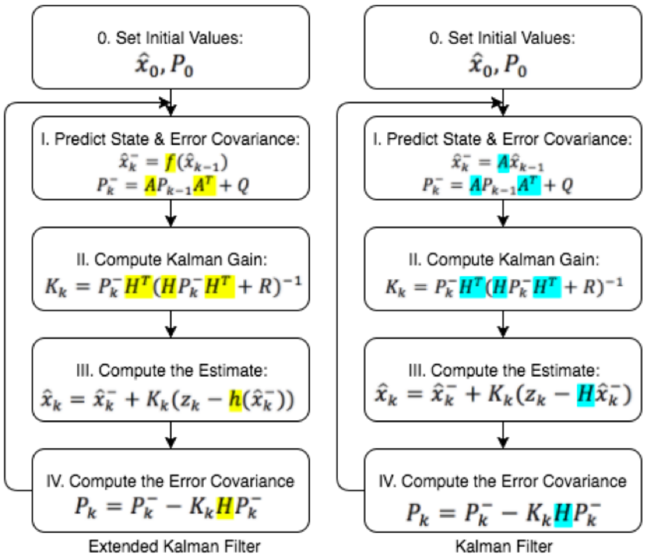

Figure 1. Diagram Comparison between EKF and KF

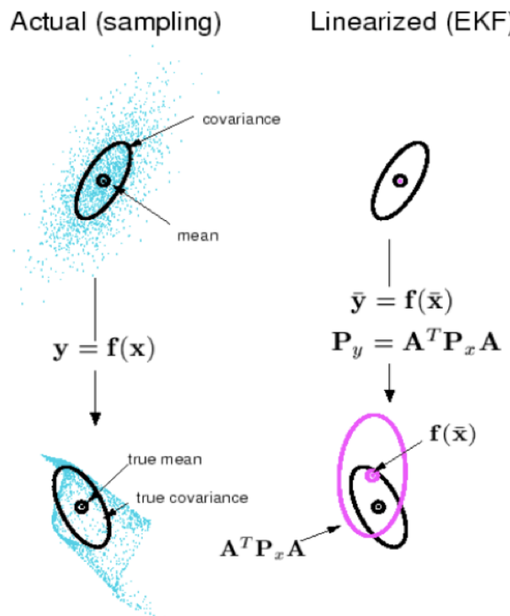

Figure 2. Mean and Covariance Propagation

The reason why matrix A is derived at the point $\hat{x}_{k-1}$ is that, compared to $\hat{x}_k^-$, the previous estimated state $\hat{x}_{k-1}$ is closer to reality so the matrix A will better approximate the actual linearized model.

The Jacobian matrix is obtained through the first order partial derivative, so the Extended Kalman Filter can be viewed as "first order" approximation to the optimal terms[6]. However, the Extended Kalman filter can introduce large errors when the high order terms cannot be overlooked. As shown in figure.2[7], linearized transformation is not reliable if propagation cannot be well approximated by a linear function[8], and actually the system may divergent. In addition, linearization can be applied only if the Jacobian matrices exists. Furthermore, the heavy computation of Jacobian matrices makes EKF impractical in many situations.

2.3. Unscented Kalman Filter

The Unscented Kalman Filter[9], proposed by Julier and Uhlman, addresses the approximation issues of the Extended Kalman Filter. The intuition of this improvement is to approximate a probability distribution rather than a nonlinear function. To achieve this, a set of points are carefully chosen to described the mean and covariance, and the points are called the sigma points[10].

The Unscented Kalman Filter can be seen as a combination of Unscented Transformation and Kalman Filter. We start first explaining the unscented transformation[11].

The unscented transformation(UT) is a method to compute the statistics of a random variable undergoing nonlinear transformation by transforming each sigma point through the nonlinear function. Suppose a random variable x (dimension n) with mean μ and covariance matrix $\sum$ is propagated through a nonlinear model with function $y = g(x)$. A matrix X of $2n+1$ vectors X_i is formed as the set of sigma points, and each vector is corresponding with weight W_i. The procedures are according to the following:

Choosing the sigma points

$$X^{[0]} = \mu \tag{8}$$

$$X^{[i]} = \mu + (\sqrt{(n+\lambda)})_i fori = 1, ..., n \tag{9}$$

$$X^{[i]} = \mu + (\sqrt{(n+\lambda)})_{i-n} fori = n+1, ..., 2n \tag{10}$$

Weights sigma points

$$w_m^{[0]} = \frac{\lambda}{n+\lambda} \tag{11}$$

$$w_c^{[0]} = w_m^{[0]} + (1 - \alpha^2 + \beta) \tag{12}$$

$$w_m^{[i]} = w_c^{[i]} = \frac{1}{2(n+\lambda)} fori = 1, ..., 2n \tag{13}$$

where $\lambda = \alpha^2(n+k) - n$ is a scaling parameter. α is usually set to small value and k is usually set to zero, and both of them determine how far the sigma points are away from the mean. β is used to incorporate prior knowledge of the distribution of x, and $\beta = 2$ is the optimal choice for Gaussians. $(\sqrt{(n+\lambda)})_i$ is the i th row of the matrix

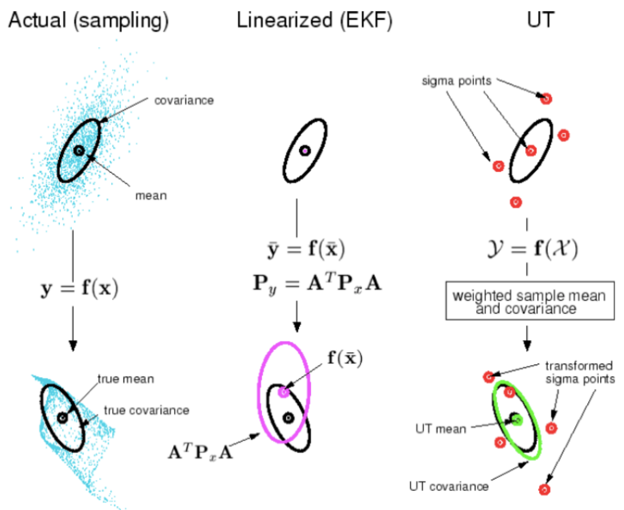

Figure 3. Mean and Covariance Propagation

square root. These sigma vectors are propagated through the nonlinear model,

$$Y_i = g(X_i) fori = 0, ..., 2n \tag{14}$$

Therefore, the mean and covariance of y can be approximated by the summation of the weighted sample mean and covariance of the transformed sigma points,

$$\mu^{'} = \sum_{i=0}^{2n} w_m^{[i]} g(X^{[i]}) \tag{15}$$

$$\sum^{'} = \sum_{i=0}^{2n} w_c^{[i]} (g(X^{[i]}) - \mu^{'})(g(X^{[i]}) - \mu^{'})^T \tag{16}$$

These sigma points completely capture the true mean and covariance accurately to the 3rd order for any nonlinearity. As shown in Figure.3[7], the right plots show the performance of the UT when using 5 sigma points. It?s obvious that the UT outperforms the other two methods, and it?s a better approximation than EKF for nonlinear models.

Therefore, when the mean and covariance of the input are certain, the mean and covariance of the non-linear transformation can be computed using UT. The algorithm of Unscented Kalman Filter is shown in Figure.4. The basic framework is the same as the Extended Kalman Filter. The difference is that, instead of using the nonlinear functions directly for prediction and the Jacobian matrices to linearize the model, the Unscented Kalman Filter applies the UT to obtain the mean and covariance of the nonlinear propagation[12], so it does not require the computation of the Jacobian matrices.

Though no Jacobian matrices are needed, the Unscented Kalman Filter belongs to the same complexity class with the

Unscented_Kalman_filter$(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)$:

$$\mathcal{X}_{t-1} = (\mu_{t-1} \quad \mu_{t-1} + \sqrt{(n+\lambda)\Sigma_{t-1}} \quad \mu_{t-1} - \sqrt{(n+\lambda)\Sigma_{t-1}})$$

$$\bar{\mathcal{X}}_t^* = g(u_t, \mathcal{X}_{t-1})$$

$$\bar{\mu}_t = \sum_{i=0}^{2n} w_m^{[i]} \bar{\mathcal{X}}_t^{*[i]}$$

$$\bar{\Sigma}_t = \sum_{i=0}^{2n} w_c^{[i]} (\bar{\mathcal{X}}_t^{*[i]} - \bar{\mu}_t)(\bar{\mathcal{X}}_t^{*[i]} - \bar{\mu}_t)^T + R_t$$

$$\bar{\mathcal{X}}_t = (\bar{\mu}_t \quad \bar{\mu}_t + \sqrt{(n+\lambda)\bar{\Sigma}_t} \quad \bar{\mu}_t - \sqrt{(n+\lambda)\bar{\Sigma}_t})$$

$$\bar{\mathcal{Z}}_t = h(\bar{\mathcal{X}}_t)$$

$$\hat{z}_t = \sum_{i=0}^{2n} w_m^{[i]} \bar{\mathcal{Z}}_t^{[i]}$$

$$S_t = \sum_{i=0}^{2n} w_c^{[i]} (\bar{\mathcal{Z}}_t^{[i]} - \hat{z}_t)(\bar{\mathcal{Z}}_t^{[i]} - \hat{z}_t)^T + Q_t$$

$$\bar{\Sigma}_t^{x,z} = \sum_{i=0}^{2n} w_c^{[i]} (\bar{\mathcal{X}}_t^{[i]} - \bar{\mu}_t)(\bar{\mathcal{Z}}_t^{[i]} - \hat{z}_t)^T$$

$$K_t = \bar{\Sigma}_t^{x,z} \, S_t^{-1}$$

$$\mu_t = \bar{\mu}_t + K_t(z_t - \hat{z}_t)$$

$$\Sigma_t = \bar{\Sigma}_t - K_t \, S_t \, K_t^T$$

return μ_t, Σ_t

Figure 4. Unscented Kalman Filter Algorithm

Extended Kalman Filter. Besides, the Unscented Kalman Filter is still restricted to Gaussian distributions.

2.4. Particle Filter

Another popular tool in object tracking is the Particle Filter[13]. The Particle Filter method, first introduced by Gordon in 1993, is an efficient method to apply simulation for estimating unknown probability distributions, and it?s also known as sequential Monte Carlo method[14]. The Particle Filter is suitable in all kinds of frameworks, regardless of whether the model is linear or whether the noise is Gaussian.

The basic idea of the algorithm is that any distribution can be represented as a weighted particle set, consisting of pairs $< x_k^{(i)}, w_k^{(i)} >$, where $x_k^{(i)}$ is a possible value of the system state and $w_k^{(i)}$ is its plausibility. Figure.5 indicates the diagram of particle filter. In particle filtering we first randomly generate a set of particle that represents the initial distribution. The posterior distribution is defined by a new particle set and it?s determined by a function f, which describes the dynamics of the system. The measurement update step is achieved by setting the weights according to the measurement data. The relation between the measurement data and the latent states is given by the conditional h. The most probably estimated state will be:

$$x_k = \sum_{i=1} x_k^{-(i)} w_k^{(i)} \tag{17}$$

It frequently occurs that almost all weights are zero ex-

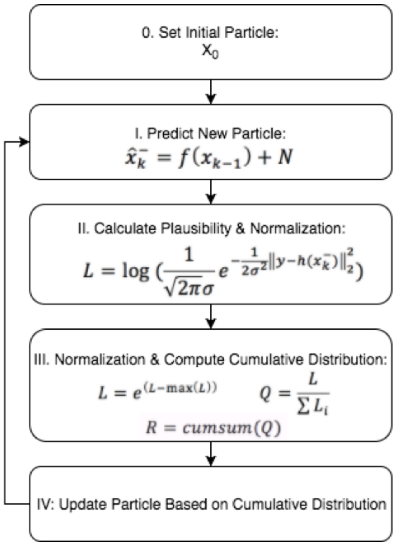

Figure 5. Diagram of Particle Filter

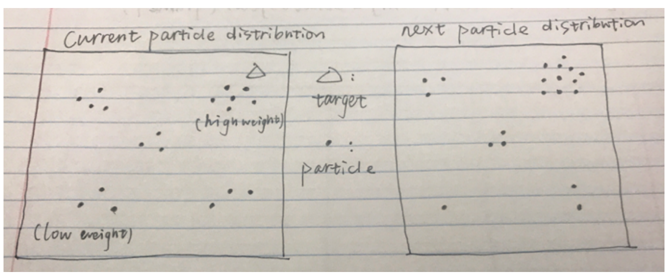

Figure 6. Particles Resampling

cept for one which tends to be one when implementing the Particle Filter in its basic form. To eliminate the effect, the resampling[15] is very important. The distribution can be updated as the particles be resampled with the weights which we obtain from the measurement data, as shown in Figure.6.

When used in object tracking, the Particle Filter is usually applied with color detection and it?s relatively easy to implement. However, since the information of a pixel is just RBG information, it cannot work for gray-scale video, and its precision decreases sharply for low-light and occluded objects. Another issue is the decision of the sample size for adequate performance. An efficient set size has been introduced by Kong and Liu[15], which can be obtained from the weights of the particles following the equation below:

$$N_{eff} = \frac{N_s}{1 + var(w_k)} \tag{18}$$

where N_s is the original set size. Notice that $N_{eff} <= N_s$, and small N_{eff} indicates severe degeneracy, which is an undesirable effect in particle filters. A direct way to reduce this effect is to use very large N_s. So sufficient amounts of particles are prerequisite to ensure accurate tracking, and it can require high computation load for large-size video.

3. Experiment Results

Experiments here

4. Conclusion

Conclusion here

References

[1] Botond A. Bocsi, Lehel Csato. Visual Tracking with Filtering Algorithms. *2008 4th International Conference on Intelligent Computer Communication and Processing*, pages: 269-274, 2008.

[2] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME?Journal of Basic Engineering*, 82(Series D): 35?45, 1960.

[3] Botond A. Bocsi, Lehel Csato. Visual Tracking with Filtering Algorithms. *M. S. Grewal and A. P. Andrews. Kalman Filtering: Theory and Practice Using MATLAB*, John Wilney and Sons, Inc., second edition, 2001.

[4] Peter S. Maybeck. Stochastic Models, Estimation, and Control. *Mathematics in Science and Engineering*, Academic Press, Inc., vol.141, 1979.

[5] H. W. Sorenson. Kalman Filtering: Theory and Application. *Piscataway, NJ: IEEE*, 1985.

[6] E. A. Wan, R. Van Der Merwe. The Unscented Kalman Filter for Nonlinear Estimation. *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications and Control Symposium*, pages: 153-158, 2000.

[7] Cyrill Stachniss. Unscented Kalman Filter.

http://ais.informatik.uni-freiburg.de/teaching/ws13/mapping/pdf/slam06-ukf-4.pdf

[8] Ye Yuan, Guiyuan Fu and Weirong Zhang. Extended and Unscented Kalman Filters for Parameter Estimation of a Hydrodynamic Model of Vessel. *2016 35th Chine Control Conference(CCC)*, pages: 2051-2056, 2016

[9] S . J. Julier and J. K. Uhlmann. Visual Tracking with Filtering Algorithms. *2Proc. of AeroSense: The 11th Int. Symp. on Aerospace/Defence Sensing, Simulation and Controls*, 1997.

[10] Ryan Turner and Carl Edward Rasmussen. Model based learning of sigma points in unscented Kalman filtering. *Machine Learning for Signal Processing(MLSP), 2010 IEEE International Workshop*, 2010.

[11] S. J. Julier and J. K. Uhlmann. Unscented filtering and nonlinear estimation. *Preceedings of the IEEE*, vol.92, no.3, pages: 401-422, 2004.

[12] Fredrik Gustafsson and Gustaf Hendeby. Some Relations Between Extended and Unscented Kalman Filters. *EEE Transactions on Signal Processing*, vol.60, no.2. pages: 545-555, 2012.

[13] N. Gordon, J. Salmond, and A. Smith. A Novel Approach to Non-linear/Non-Gaussian Bayesian State Estimation. *IEEE Preceedings on Radar and Signap Processing*, pages: 107? 113, 1993.

[14] C. P. Robert and G. Casella. Monte Carlo Methods. *Springer*, second edition, 2004.

[15] A. Doucet, S. Godsill, and C. Andrieu. On Sequential Monte Carlo Sampling Methods for Bayesian filtering. *Statistics and Computing*, vol.10, no.3, pages:197?208, 2005.