Target Tracking Using Kalman Filter Based Algorithms

Jiankun Ling*
College of Engineering, South China Agricultural University, Guangzhou, Guangdong, 510000, China
*Corresponding author’s e-mail: 201821120211@stu.scau.edu.cn

Abstract. Kalman filter and its families have played an important role in information gathering, such as target tracking. Data association techniques have also been developed to allow the Kalman filter to track multiple targets simultaneously. This paper revisits the principle and applications of the Kalman filter for single target tracking and multiple hypothesis tracking (MHT) for multiple target tracking. We present the brief review of the Bayes filter family and introduce a brief derivation of the Kalman filter and MHT. We show examples for both single and multiple targets tracking in simulation to illustrate the efficacy of Kalman filter-based algorithms in target tracking scenarios.

1. Introduction
Target state estimation, or tracking, is an essential technique over decades that aims to obtain continuous or discrete states of one or more desired objects from sensor measurements and recursively update them over time. Target states may vary from geometric information such as position and orientation to any other information one may interested in, such as color and battery level, resulting in a broad class of applications ranging from robot localization and mapping to surveillance and autonomous driving [1, 2]. Probabilistic manners are often applied in target state estimations due to noise and uncertainty in the real world. Such methods represent the uncertain target states via the belief distributions as time steps move forward. The most general algorithm for calculating beliefs is given by the Bayes filter algorithm, which calculates the belief distribution from measurement and control data, with a Markov assumption that the current state includes complete information of the past [3]. However, in practice, the exact belief distributions are often intractable in continuous spaces due to computational complexity, and approximation techniques are required in most robotics problems. In general, two classes of approximation skills have been developed, i.e., the parametric techniques and the non-parametric techniques.

The most important and popular family of parametric state estimators are Gaussian filters, which represent the beliefs by multivariate Gaussian distributions. For a linear Gaussian system, a Kalman filter (KF) can be applied to represent the beliefs by the moment representation, i.e., the mean and the covariance in a way that yields to Gaussian posterior beliefs propagated recursively with closed-form solutions [4]. The extended Kalman filters (EKF) release the linearity assumptions by expressing the next state probabilities and the measurement probabilities using nonlinear functions and calculates an approximation to true belief represented by a Gaussian [5]. As noted by Wan et al., the EKF allows large errors in the true posterior, which may result in sub-optimal performance and even divergence [6]. The unscented Kalman filter (UKF) was later proposed to overcome the flaws in using EKF by representing the state distributions via sample points [7]. All above mentioned Gaussian filters can track the beliefs in polynomial time.
Non-parametric filters approximate beliefs by a finite number of values rather than a fixed functional form. As a result, non-parametric filters can track both continuous and discrete states with adaptive computational time depending on the approximation accuracy. One example is the histogram filter, which decomposes the state space into many finite regions and represents the cumulative posterior for each region by a single probability value [3]. Another non-parametric implementation of the Bayes filter named the particle filter represents the posterior by a random scatter of samples drawn from this posterior [8]. The particle filter approximates posterior over a particular region via weighted particles in that region by Monte-Carlo-based statistical signal processing. Compared with the most popular Kalman filter, the particle filter has a lower system requirement, i.e., neither the system model equation is required to be linear, nor the system noise is required to be Gaussian and better scalability and universality. However, many samples may be required to approximate the posterior accurately, which brings a heavier computational burden.

For the state estimators to track multiple discrete targets, data association is often required to match estimated target tracks to true targets. Many multiple target tracking (MTT) techniques have been developed over the years, such as global nearest neighbor (GNN), joint probabilistic data association (JPDA), multiple hypothesis tracking (MHT), and particle filters based method. Each of them solves the data association problem differently [9-12]. Such multiple target trackers figure out a unique label for each track (target) over time. Another class of multiple target trackers, based on random finite set statistics, does not require explicit data association [13]. For example, the probability hypothesis density (Ph.D.) filter represents the multiple targets posterior via the first-order moment (mean) of target density distribution and propagate it over time. In this way, the environment and measurement noise is incorporated in the form of the intensity of target distribution [14]. It is especially suitable in scenarios where unique target labels are not required, such as rescuing people from fire scenes.

In this paper, our contributions are three-fold. Firstly, we briefly investigate the background of the generic Bayes filter and its mathematic derivation. Secondly, we revisit the Kalman filter and the multiple hypothesis filter in detail under single target tracking and multiple target tracking scenarios, respectively. Thirdly, we show simulation demos and results for single and multiple target tracking using our introduced trackers and analyze the efficacy.

2. Bayes Filter

In this paper, a Bayes filter is tasked to recursively estimate the position of a set of n targets in a 2-D plane using sensor measurement \( Z = \{z_1, z_2, ..., z_n\} \) where \( n = 1, 2, ..., \) and \( Z \subseteq \mathbb{R}^2 \). The true set of positions at each time step \( k \) is denoted by \( X = \{x_1, x_2, ..., x_n\} \) where \( X \subseteq \mathbb{R}^2 \). A single target tracking problem will be formulated when \( n = 1 \) a multiple target tracking problem will be formulated otherwise. In practice, the measurements can be obtained by either a fixed or a mobile sensor and by either one or more sensors depending on the applications. Typical sensor models that can measure target planet positions include the camera, laser scan, radar, etc.

The Bayes filtering technique only calculates the approximate state posterior of the current moment, which will then be utilized in the next discrete time step for state posterior estimation. This is an iterative process based on the Markov assumption that the current estimated state \( X_i \) is a complete summary of all past estimations \( \{x_{i-1}, ..., x_0\} \). The filter can estimate the state \( X_{i+1} \) the next time without relying on previous information.

The framework of Bayes filtering is composed of two steps: the prediction step and the update (correction) step. The prediction step takes in the state transition probability, i.e., the probability of a target with the state \( X_{i-1} \) at time step \( k-1 \) transit to state \( x_i \) at a time step \( k \), and predicts the next state of posteriors. The update step incorporates the measurement model utilizing the sensor measurement to update (correct) the previous prediction. In this paper, the transition probability takes the form of the
target motion model, which can be obtained before the task using data-driven manners or based on experiences.

Based on the Markov assumption given by

One-section Markov:

\[ p(x_k | x_{k-1}, z_{k-1}) = p(x_k | x_{k-1}) \]  

(1)

One-section Markov:

\[ p(x_{k-1} | x_{k-1}, z_{k-1}) = p(x_{k-1} | x_{k-1}) \]  

(2)

Conditional independence:

\[ p(z_k | x_{k}, z_{k-1}) = p(z_k | x_k) \]  

(3)

The Bayes filter prediction step is outlined by

\[ p(x_{k-1} | z_{k-1}) \rightarrow p(x_k | z_{k-1}) \]  

(4)

\[ p(x_k | z_{k-1}) = \int p(x_k | x_{k-1})p(x_{k-1} | z_{k-1})dx_{k-1} , \]  

(5)

and the update step is given by

\[ p(x_k | z_{k}) \rightarrow p(x_k | z_k) \]  

(6)

\[ p(x_k | z_k) = \frac{1}{Z_k} p(x_k | z_k)p(x_k | z_{k-1}) \]  

(7)

\[ Z_k = \int p(x_k | z_k)p(x_k | z_{k-1})dx_k . \]  

(8)

3. Kalman Filter for Single Target Tracking

The Kalman filter represents the target state posterior in the form of Gaussian distribution and recursively estimates the mean and covariance of it. As a Gaussian implementation of the Bayes filter, the main components of the Kalman filter are to predict the current state using a linear target transaction model and to update the prediction based on the sensor measurement and the linear control input (if any).

The one-step prediction of the state vector is given by

\[ \hat{x}(k+1|k) = F(k)\hat{x}(k|k) . \]  

(9)

The predicted error covariance matrix is given by

\[ P(k+1|k) = F(k)P(k|k)F^T(k) + Q(k) . \]  

(10)

Using the measured value \( Z(k+1) \) at time \( k + 1 \), the innovation or residual is obtained as

\[ u(k+1) = Z(k+1) - H(k+1)\hat{x}(k+1|k) . \]  

(11)

The innovation covariance matrix is

\[ S(k+1) = H(k+1)P(k+1|k)H^T(k+1) + R(k+1) . \]  

(12)

Kalman gain is

\[ K(k+1) = P(k+1|k)H^T(k+1)S(k+1)^{-1} . \]  

(13)
Use innovation to correct the predicted value of the state vector and update the state vector to
\[
\hat{X}(k+1|k+1) = \hat{X}(k+1|k) + K(k+1)j(k+1) .
\] (14)

The state error covariance matrix is updated to
\[
P(k+1|k+1) = (I - K(k+1)H(k+1))P(k+1|k) .
\] (15)

From a filtering point of view, \(\hat{X}(k+1|k)\) and \(P(k+1|k)\) are the best one-step predicted value and the covariance matrix of the predicted signal, respectively. \(K(k+1)\) is the best filter gain, \(\hat{X}(k+1|k+1)\) and \(P(k+1|k+1)\) are the best-filtered value and the smoothed estimation covariance matrix, respectively.

Under ideal conditions, the filtered value \(\hat{X}(k+1|k+1)\) obtained by the above recursive algorithm is the linear unbiased minimum variance estimate of state \(X(k+1)\), and \(P(k+1|k+1)\) is the minimum mean square error matrix. Kalman filter can obtain the state's estimated value when performing filtering processing and perform error analysis on the obtained state error covariance, which is an important advantage of it.

4. Multiple Hypothesis Tracking for Multiple Target Tracking

The MHT algorithm builds a potential tracking hypothesis tree for each candidate target, calculates the belief of each track, and selects the most likely tracking combination. Since the entire tracking assumption is considered when calculating the probability, MHT is theoretically suitable for higher-order information such as long-term motion and appearance models. The MHT considers miss detections and target birth and death and assumes that the number and initial states of targets are unknown.

4.1. Tracking Tree Maintaining and Gain Gating

The first step of MHT is to create and update a tracking tree composed of a sequence of the time frame that holds multiple hypotheses generated by observations of targets. In each frame, a new tracking tree will be generated as a new target being detected. Meanwhile, an existing tracking tree will be updated if an existing target maintains being detected.

A gating step will then follow if the tracking hypothesis is within the observed gating range, meaning that a new observation is close to an existing track (determined by a distance \(d\)) and is very likely to belong to it. The gating step predicts the location where the next observation is likely to appear to reduce the number of data association pairs. Elliptical gating is one of the most common gating techniques. We introduce the rectangular gating in this paper since it is more computationally efficient. The observation corresponding to the increasing branch of the tree is given by
\[
d^2 = (\hat{x}_k^i - y_i)^T \left( \sum_{i}^{-1} \right) (\hat{x}_k^i - y_i) \leq d_{th}
\] (16)

where the distance threshold \(d_{th}\) determines the size of the gating area and \(x_k^i\) denotes the possible position of the \(i\) track at a time \(k\), obtained by the Kalman filter. The rectangular gating uses the Mahalanobis distance to decide whether to use the newly predicted target to update a specific trajectory.
4.2. Best Global Hypothesis

Only the best global hypothesis is retained by calculating the tracking score, and the others are not. The \( i_{th} \) track’s score at frame \( k \) is defined as follows:

\[
S'_{i}(k) = \omega_{mot} S'_{mot}(k) + \omega_{app} S'_{app}(k)
\]  

(17)

where \( S'_{mot}(k) \) and \( S'_{app}(k) \) are the motion and appearance scores, and \( \omega_{mot}, \omega_{app} \) are the weights that control the contribution of the location measurement \( y_{i_k} \) and the appearance measurement \( X_{i_k} \) to the track score, respectively. \( S'_{mot}(k) \) and \( S'_{app}(k) \) are defined by

\[
S'_{mot}(k) = \ln \frac{p\left(y_{i_k} | i_{k} \leq T_{i}\right)}{p\left(y_{i_k} | i_{k} \leq \phi\right)}
\]  

(18)

\[
S'_{app}(k) = \ln \frac{p\left(X_{i_k} | i_{k} \leq T_{i}\right)}{p\left(X_{i_k} | i_{k} \leq \phi\right)} = \ln \frac{p\left(i_{k} \leq T | X_{i_k}\right)}{p\left(i_{k} \leq \phi | X_{i_k}\right)}.
\]  

(19)

If \( S'(k) > 0 \), a tracking hypothesis is more like a true target, while \( S'(k) < 0 \) means that it is more likely to be a false warning. So we can recursively calculate the best global hypothesis by

\[
S'(k) = S'(k-1) + \Delta S'(k),
\]  

(20)

\[
\Delta S'(k) = \begin{cases} 
\ln \frac{1-P_{D}}{1-P_{F}} \approx \ln(1-P_{D}), & \text{if } i_{k} = 0 \\
\omega_{mot} \Delta S'_{mot}(k) + \omega_{app} \Delta S'_{app}(k), & \text{otherwise}
\end{cases}
\]  

(21)

where \( \Delta S'_{mot}(k) \) and \( \Delta S'_{app}(k) \) represent the increments of the two types of scores at the time \( k \), respectively, \( P_{D} \) and \( P_{F} \) represent the detection probability and false alarm probability, respectively.

4.3. Track Tree Cropping

The \( N \)-scan pruning method is one of the choices following the previous steps to control the number of hypotheses by limiting the depth of the trajectory tree. The \( N \)-scan pruning method forces the uncertainty generated at \( N \) the time to be resolved at \( k+N \) the time. When the depth of the trajectory tree is greater than \( N \), the \( N \)-scan pruning method will search for the leaf node with the highest confidence in the trajectory tree and keep the root node branch of the leaf node with the highest confidence, then delete the remaining branches. We set the threshold \( B_{th} \) such that when the number of branches exceeds \( B_{th} \), only the first \( B_{th} \) branches are retained based on the tracking score.

For each tracking hypothesis, a method of online training appearance model is introduced. The appearance model can be effectively learned by the least squares method of regularization, and only a little extra operation is required for each hypothesis branch.
In addition, when the appearance model score \( F(X_i) < c_2 \), the setting \( \Delta S_{app}(t) = -\infty \) can prevent pruning. This can reduce unnecessary pruning and improve efficiency.

5. Examples
We conduct simulation using MATLAB to present examples both for single and multiple target tracking. For both examples, we track targets from .mp4 videos which show fixed scenes with one or more objects moving within the scenes. We use function kalmanFilterForTracking() for single target tracking and function MotionBasedMultiObjectTrackingExample() for multiple target tracking, both in computer vision toolbox.

5.1 Single Target Tracking
We select to use the MATLAB computer vision toolbox video where a small ball moves from left to right on the floor, passing through a carton. Because of the friction on the ground, we choose a constant acceleration motion model. A paper box is set up in the middle of the ball's trajectory to verify whether the Kalman filter is still able to track the trajectory of the ball by the target motion model in the prediction step when the ball is unable to be detected. We set the initial position input to the position where the object was first detected. We also set the initial estimation error vector to a larger value because the initial state can be very noisy. After all, it is derived from a single detection. We select the value of measurement noise according to the accuracy of the camera. Setting the measurement noise to a larger value will result in a less accurate detector.

The result is shown in Fig.1a. We can see that the ball is tracked throughout its trajectory correctly, even with the paper box blocking at the halfway. This proves that the Kalman filter can predict the target state using the motion model and recursively correct it over time. The tracking can recover with the prediction step even no detection is received for a while. It embodies the superiority of the Kalman filter algorithm in state estimation and accuracy for linear systems. Fig.1b presents a failed example of tracking a moving ball recorded in a video. The ball is no longer tracked as it's blocked from detected by an obstacle. This is due to the inaccuracy of the motion model and the confusion of extracting objects from the background with a similar color in the video. And shows that while using an inaccurate motion model. The object appears in a completely different position from the predicted position. When the object is released, it undergoes a constant deceleration due to resistance from the ground. Therefore, the constant acceleration model is a better choice. If the optimal motion model is not appropriate, the tracking results will be sub-optimal. Meanwhile, when there is a problem with the initial position we set, the initial center position of the object will deviate. When we give the initial estimation a significant error, the object's trajectory will find a serious deviation from being blocked by the obstacle to the end of the trajectory. This indicates that for the Kalman filter, a suitable parameter configuration plays an important role.

In short, the multi-target tracking we experimented with is mainly divided into two steps: Detect moving target in every frame and associating the detected target with the same target that was being tracked before. And our first step is divided into using the Gaussian mixture model for background subtraction to get the moving target, eliminating the noise in the foreground mask through morphological operations, and detecting the connected domain through blob analysis to obtain the corresponding moving target. Our second step is divided into using Kalman filter to calculate and predict the position of each trajectory in the next frame. Then calculate the Euclidean distance between the predicted trajectory position and each newly detected target, and measure The result is used as the loss function matrix. Then, the Hungarian matching algorithm is used to calculate the assigned/unassigned tracking/detection according to the threshold. And update tracking and detection, update assigned tracking to the assigned detection position of the current frame, delete unassigned tracking that reaches the threshold (continuous disappearance of multiple frames) and add unassigned detection to the track.
5.2 Multiple Target Tracking

We test multiple target tracking using an MHT-based algorithm in two scenes. The first one is on a flat road, with few people and simple background. Another scene is on a slope, with a more complicated environment and more people. In the two videos, we aim to track people, and most of them are moving at a constant speed. We make a comparison in this test between simple and complex scenarios.

The multi-target tracking includes mainly two steps: detect moving target in every frame and associating the detected target with the same target that was being tracked before. Our first step is divided into using the Gaussian mixture model for background subtraction to get the moving target, eliminating the noise in the foreground mask through morphological operations, and detecting the connected domain through blob analysis to obtain the corresponding moving target. Our second step is divided into using the Kalman filter to calculate and predict the position of each trajectory in the next frame and calculating the Euclidean distance between the predicted trajectory position and each newly detected target. The result is evaluated with the loss function matrix, and then the Hungarian matching algorithm is used to calculate the assigned/unassigned tracking/detection according to the threshold. Finally, we update assigned tracking to the assigned detection position of the current frame, delete unassigned tracking that reaches the threshold (continuous disappearance of multiple frames) and add unassigned detection to the track.

Fig.2a and Fig.2b demonstrate the results of tracking multiple people in videos. Most people are successfully tracked in both simple and complex environments even within a short appearance time and been marked with rectangles separately. This reflects that the MHT algorithm can track targets fast and reliably using Kalman filter with correct motion model characterization and match tracked target to new measurements as time moves forward.
Figure 2. Figures show the results of the recognition of people in two environments under multi-target tracking, and the yellow rectangle is the recognition of the target contour.

We also conduct some extra tests where the tracking of people is not very accurate, which will be more obvious in a complex environment compared to a simple environment. Fig. 2d shows that two people in front of the street sign in the lower right corner are not detected. This may be because they lose detection as they are relatively small and standing still, which is not described by our selected motion model. In Fig. 2c, two people are detected as one merged target. The reason is that the morphological operation and connectivity check are not too detailed. In Fig. 2e, the man in white clothes in the lower left corner does not appear on the right edge because it has not reached the threshold of the minimum number of consecutive frames. The person in the far lower left of the picture is not detected because the object is too small, and the movement is very slow.

These examples indicate that the algorithm has limitations as this article is based on dynamic multi-target tracking. Thus, continuous multiple frames of static objects will be automatically classified as background. We also find that moving vehicles may block people from being tracked. To increase the reliability in using MHT in real-world scenarios, the morphological operation and connectivity check should be more detailed to be better divided. Meanwhile, for different objects, different eigenvalues should be given, and some methods should be adopted. It is used to divide different objects in foreground detection and only track the objects we are interested in.

6. Conclusion
This paper introduces the Kalman filter algorithm for single target tracking and the multiple hypothesis tracking (MHT), a Kalman filter-based algorithm incorporated with data association technique for multiple target tracking. Kalman filter predicts the mean and covariance of the Gaussian target state posterior using the target transition model and updates the sensor measurement recursively. MHT uses the Kalman filter to estimate the states of targets at each step and matches measurements to target tracks with a series of data association processes, including maintaining tracking trees, gating, and pruning. Experiments validate the efficacy of single and multiple target tracking using Kalman filter and MHT, respectively. It is also concluded that the morphological operation and connectivity check affect the tracking effect. The different feature values assigned to different objects may lead to various performances in detecting and tracking objects.

References
[1] Se, Stephen, David Lowe, and Jim Little. "Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks." The international Journal of robotics Research 21.8 (2002): 735-758.
[2] Levinson, J., Askeland, J., Becker, J., Dolson, J., Held, D., Kammel, S., Kolter, J.Z., Langer, D., Pink, O., Pratt, V. and Sokolsky, M., (2011). Towards fully autonomous driving: Systems and algorithms. In 2011 IEEE Intelligent Vehicles Symposium (IV) (pp. 163-168). IEEE.
[3] Thrun, S., (2002). Probabilistic robotics. Communications of the ACM.
[4] Welch, G., & Bishop, G. (1995). An introduction to the Kalman filter.
[5] Hoshiya, M. and Saito, E., (1984). Structural identification by extended Kalman filter. Journal of engineering mechanics, 110(12), pp.1757-1770.

[6] Wan, E.A. and Van Der Merwe, R., 2000, October. The unscented Kalman filter for nonlinear estimation. In Proceedings of the IEEE (2000) Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373) (pp. 153-158). Ieee.

[7] Julier, S.J. and Uhlmann, J.K., 1997, July. New extension of the Kalman filter to nonlinear systems. In Signal processing, sensor fusion, and target recognition VI (Vol. 3068, pp. 182-193). International Society for Optics and Photonics.

[8] Carpenter, J., Clifford, P. and Fearnhead, P., (1999). Improved particle filter for nonlinear problems. IEE Proceedings-Radar, Sonar and Navigation, 146(1), pp.2-7.

[9] Konstantinova, P., Udvarov, A. and Semerdjiev, T., (2003), June. A study of a target tracking algorithm using global nearest neighbor approach. In Proceedings of the International Conference on Computer Systems and Technologies (CompSysTech’03) (pp. 290-295).

[10] Fortmann, T., Bar-Shalom, Y. and Scheffe, M., (1983). Sonar tracking of multiple targets using joint probabilistic data association. IEEE journal of Oceanic Engineering, 8(3), pp.173-184.

[11] Blackman, S.S., (2004). Multiple hypothesis tracking for multiple target tracking. IEEE Aerospace and Electronic Systems Magazine, 19(1), pp.5-18.

[12] Hue, C., Le Cadre, J.P. and Pérez, P., (2002). Sequential Monte Carlo methods for multiple target tracking and data fusion. IEEE Transactions on signal processing, 50(2), pp.309-325.

[13] Vo, B.T., 2008. Random finite sets in multi-object filtering (p. 254). University of Western Australia.

[14] Vo, B.N. and Ma, W.K., 2006. The Gaussian mixture probability hypothesis density filter. IEEE Transactions on signal processing, 54(11), pp.4091-4104.