Recursive Bayesian Tracking

Kuan-Lin Chen*
UC San Diego
kuc029@ucsd.edu

Hsiao-Chen Huang†
UC San Diego
hsh030@eng.ucsd.edu

Emal Fatima‡
UC San Diego
efatima@eng.ucsd.edu

Abstract

Bayesian filtering provides a general framework for propagating probability density functions. In this paper, both probabilistic reasoning and state-space representation for Bayesian filtering are gone through rigorous derivation and linked together. We carefully derive the Bayesian solutions under the Gaussian assumption from two different perspectives, i.e., linear minimum mean squared error (LMMSE) and Bayes' theorem. In this case, the optimal solution of Bayesian filtering is linear Kalman filter. For those physical systems that do not obey Gaussian assumption, we only have suboptimal solutions such as sigma point class Kalman filters, Monte Carlo Kalman filter and sequential importance sampling (SIS) particle filters. We apply our proposed algorithms to real-world tracking problems, and demonstrate their performance on a standard benchmark. The experimental results show that the performance can be dramatically boosted by incorporating prior correctly in the Bayesian framework. Additionally, all parameters and statistics are updated in a recursive manner which allows online data processing as well as rapid adaptation to changing signals.

1. Recursive Bayesian Tracking

Conceptually, many problems in science require recursive estimation of the state $\mathbf{x}_n \in \mathbb{R}^{n_x}$ of a system that changes over time n using a sequence of noisy measurements $\{\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_n \in \mathbb{R}^{n_z}\} \triangleq \mathbf{z}_{1:n}$ made on the system [1]. For example, in the application of object tracking, the state at time n can be the descriptor of the bounding box of a object given by

$$\mathbf{x}_n = \begin{bmatrix} x_n \\ y_n \\ w_n \\ h_n \end{bmatrix} \in \mathbb{R}^4 \tag{1}$$

where (x_n, y_n) is the Cartesian coordinate, and w_n and h_n are the width and height of the bounding box respectively at time n. A measurement $\mathbf{z}_n$ can be obtained by making a transformation on the frame at time n. An illustration of multiple object tracking is given by Fig. 1. Given a set of

Figure 1: An illustration of multiple object tracking [9] and our target of interests.

measurements $\{\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_p \in \mathbb{R}^{n_z}\} \triangleq \mathbf{z}_{1:p}$, our goal is to find the conditional posterior density $p(\mathbf{x}_n|\mathbf{z}_{1:p})$ to estimate the state $\mathbf{x}_n$ of a system at time n to the best of our knowledge using the measurements up to time p. In the sense of minimum mean squared error (MMSE), the estimate $\hat{\mathbf{x}}_{n|p}$ of the state is then given by

$$\begin{aligned} \hat{\mathbf{x}}_{n|p} &= \mathbb{E}\left[\mathbf{x}_n|\mathbf{z}_{1:p}\right] \\ &= \int \mathbf{x}_n p(\mathbf{x}_n|\mathbf{z}_{1:p}) d\mathbf{x}_n \end{aligned} \tag{2}$$

which is simply a mean of $\mathbf{x}_n$ given measurements $\mathbf{z}_{1:p}$. Accordingly, the uncertainty measure of the estimate, i.e., covariance of the state is given by

$$\mathbf{P}^{\mathbf{xx}}_{n|p} = \mathbb{E}\left[(\mathbf{x}_n - \hat{\mathbf{x}}_{n|p})(\mathbf{x}_n - \hat{\mathbf{x}}_{n|p})^T|\mathbf{z}_{1:p}\right] \tag{3}$$

For many applications, we are required to develop a causal system, so we have $p \leq n$ in general.

*Kuan-Lin Chen contributed 50% of the project.
†Hsiao-Chen Huang contributed 35% of the project.
‡Emal Fatima contributed 15% of the project.

1.1. Recursive Bayesian Filtering of Probability Density Functions

Given the previous posterior $p(\mathbf{x}_{n-1}|\mathbf{z}_{1:n-1})$, the link between the current posterior $p(\mathbf{x}_n|\mathbf{z}_{1:n})$ and the previous posterior $p(\mathbf{x}_{n-1}|\mathbf{z}_{1:n-1})$ can be obtained by using Bayes' theorem and Chapman-Kolmogorov euqation. In the first step, by Bayes' theorem, we have

$$\begin{aligned}p(\mathbf{x}_n|\mathbf{z}_{1:n}) &= \frac{p(\mathbf{z}_{1:n}|\mathbf{x}_n)p(\mathbf{x}_n)}{p(\mathbf{z}_{1:n})} \\ &= \frac{p(\mathbf{z}_n,\mathbf{z}_{1:n-1}|\mathbf{x}_n)p(\mathbf{x}_n)}{p(\mathbf{z}_n,\mathbf{z}_{1:n-1})} \\ &= \frac{p(\mathbf{z}_n|\mathbf{z}_{1:n-1},\mathbf{x}_n)p(\mathbf{z}_{1:n-1}|\mathbf{x}_n)p(\mathbf{x}_n)}{p(\mathbf{z}_n|\mathbf{z}_{1:n-1})p(\mathbf{z}_{1:n-1})} \\ &= \frac{p(\mathbf{z}_n|\mathbf{z}_{1:n-1},\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{z}_{1:n-1})p(\mathbf{z}_{1:n-1})p(\mathbf{x}_n)}{p(\mathbf{z}_n|\mathbf{z}_{1:n-1})p(\mathbf{z}_{1:n-1})p(\mathbf{x}_n)} \\ &= \frac{p(\mathbf{z}_n|\mathbf{z}_{1:n-1},\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{z}_{1:n-1})}{p(\mathbf{z}_n|\mathbf{z}_{1:n-1})} \\ &= \frac{p(\mathbf{z}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{z}_{1:n-1})}{p(\mathbf{z}_n|\mathbf{z}_{1:n-1})}\end{aligned} \tag{4}$$

Since we need the prior state density $p(\mathbf{x}_n|\mathbf{z}_{1:n-1})$, one last step is needed to create a completely recursive form for the conditional probability density or current posterior $p(\mathbf{x}_n|\mathbf{z}_{1:n})$. Assume that $\{\mathbf{x}_n\}$ is a Markov process of order 1, the Chapman-Kolmogorov equation provides the link given by

$$\begin{aligned}p(\mathbf{x}_n|\mathbf{z}_{1:n-1}) &= \int p(\mathbf{x}_n|\mathbf{x}_{n-1},\mathbf{z}_{1:n-1})p(\mathbf{x}_{n-1}|\mathbf{z}_{1:n-1})d\mathbf{x}_{n-1} \\ &= \int p(\mathbf{x}_n|\mathbf{x}_{n-1})p(\mathbf{x}_{n-1}|\mathbf{z}_{1:n-1})d\mathbf{x}_{n-1}\end{aligned} \tag{5}$$

Notice that the season why we can replace $p(\mathbf{x}_n|\mathbf{x}_{n-1},\mathbf{z}_{1:n-1})$ by $p(\mathbf{x}_n|\mathbf{x}_{n-1})$ is completely based on the Markovian assumption for the process $\{\mathbf{x}_n\}$. Finally, for the normalizing constant $p(\mathbf{z}_n|\mathbf{z}_{1:n-1})$, by using marginalization and the prior state density $p(\mathbf{x}_n|\mathbf{z}_{1:n-1})$ we have developed in (5), we have

$$\begin{aligned}p(\mathbf{z}_n|\mathbf{z}_{1:n-1}) &= \int p(\mathbf{z}_n|\mathbf{x}_n,\mathbf{z}_{1:n-1})p(\mathbf{x}_n|\mathbf{z}_{1:n-1})d\mathbf{x}_n \\ &= \int p(\mathbf{z}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{z}_{1:n-1})d\mathbf{x}_n\end{aligned} \tag{6}$$

Hence, a recursive link has been established between the previous posterior $p(\mathbf{x}_{n-1}|\mathbf{z}_{1:n-1})$ and the current posterior $p(\mathbf{x}_n|\mathbf{z}_{1:n})$ that requires specification of transition density $p(\mathbf{x}_n|\mathbf{x}_{n-1})$ and likelihood function $p(\mathbf{z}_n|\mathbf{x}_n)$.

Apart from directly specifying the distributions to transition density $p(\mathbf{x}_n|\mathbf{x}_{n-1})$ and likelihood function $p(\mathbf{z}_n|\mathbf{x}_n)$, we can actually describe those dynamics under the Markovian model by using state evolution equation and measurement equation which compose a state-space representation.

1.2. State-space Representation

A discrete-time and time-varying state-space system can be described by state evolution equation

$$\mathbf{x}_n = \mathbf{f}_{n-1}(\mathbf{x}_{n-1},\mathbf{v}_{n-1}) \tag{7}$$

and measurement equation

$$\mathbf{z}_n = \mathbf{h}_n(\mathbf{x}_n,\mathbf{w}_n) \tag{8}$$

where $\mathbf{x}_n \in \mathbb{R}^{n_x}$ and $\mathbf{z}_n \in \mathbb{R}^{n_z}$ are the state and measurement of the system at time $n \geq 1$. $\mathbf{f}_{n-1} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_v} \to \mathbb{R}^{n_x}$ and $\mathbf{h}_n : \mathbb{R}^{n_x} \times \mathbb{R}^{n_w} \to \mathbb{R}^{n_z}$ are deterministic state transition and measurement functions. $\mathbf{v}_{n-1}$ and $\mathbf{w}_n$ are process and measurement noises. n_x, n_z, n_v and n_w are dimensions of the random vectors.

For the state process $\{\mathbf{x}_n\}$ to be Markovian, the following assumptions are required. All $\mathbf{v}_{n-1}$ are independent of each other and the initial state $\mathbf{x}_0$. For the state-space system to proceed, we need to know state dynamics $\mathbf{f}_{n-1}$, state noise $\mathbf{v}_{n-1}$, measurement dynamics $\mathbf{h}_n$, measurement noise $\mathbf{w}_n$, and initial state $\mathbf{x}_0$ before tracking. Hence, parameter learning techniques such as gradient descent [8] or expectation maximization (EM) [3] are required.

1.3. Solving Density-Weighted Integrals

Let us consider the first and second order statistics of the prior state density. According to (2), (5) and (7), we have

$$\begin{aligned}\hat{\mathbf{x}}_{n|n-1} &= \mathbb{E}\left[\mathbf{x}_n|\mathbf{z}_{1:n-1}\right] \\ &= \int \mathbf{x}_n p(\mathbf{x}_n|\mathbf{z}_{1:n-1})d\mathbf{x}_n \\ &= \int\int \mathbf{x}_n p(\mathbf{x}_n|\mathbf{x}_{n-1})p(\mathbf{x}_{n-1}|\mathbf{z}_{1:n-1})d\mathbf{x}_{n-1}d\mathbf{x}_n \\ &= \int\int \mathbf{f}_{n-1}(\mathbf{x}_{n-1},\mathbf{v}_{n-1})p(\mathbf{x}_n|\mathbf{x}_{n-1})p(\mathbf{x}_{n-1}|\mathbf{z}_{1:n-1})d\mathbf{x}_{n-1}d\mathbf{x}_n\end{aligned} \tag{9}$$

$$\begin{aligned}\hat{\mathbf{P}}^{\mathbf{xx}}_{n|n-1} &= \mathbb{E}\left[(\mathbf{x}_n-\hat{\mathbf{x}}_{n|n-1})(\mathbf{x}_n-\hat{\mathbf{x}}_{n|n-1})^T|\mathbf{z}_{1:n-1}\right] \\ &= \int (\mathbf{x}_n-\hat{\mathbf{x}}_{n|n-1})(\mathbf{x}_n-\hat{\mathbf{x}}_{n|n-1})^T p(\mathbf{x}_n|\mathbf{z}_{1:n-1})d\mathbf{x}_n \\ &= \int\int (\mathbf{x}_n-\hat{\mathbf{x}}_{n|n-1})(\mathbf{x}_n-\hat{\mathbf{x}}_{n|n-1})^T \\ &\quad p(\mathbf{x}_n|\mathbf{x}_{n-1})p(\mathbf{x}_{n-1}|\mathbf{z}_{1:n-1})d\mathbf{x}_n d\mathbf{x}_{n-1} \\ &= \int\int (\mathbf{f}_{n-1}(\mathbf{x}_{n-1},\mathbf{v}_{n-1})-\hat{\mathbf{x}}_{n|n-1})(\mathbf{f}_{n-1}(\mathbf{x}_{n-1},\mathbf{v}_{n-1})-\hat{\mathbf{x}}_{n|n-1})^T \\ &\quad p(\mathbf{x}_n|\mathbf{x}_{n-1})p(\mathbf{x}_{n-1}|\mathbf{z}_{1:n-1})d\mathbf{x}_n d\mathbf{x}_{n-1}\end{aligned} \tag{10}$$

In general, we may not have closed-form expressions for the density-weighted integrals in (9) and (10). Assumptions are made to circumvent the difficulties.

1.4. Hierarchy of Bayesian Trackers

Therefore, different families of functions $\mathbf{f}_{n-1}, \mathbf{h}_n$ and noises $\mathbf{v}_{n-1}, \mathbf{w}_n$ lead to different numerical methods for evaluating density-weighted integrals in (9) and (10). Thus there are two main assumptions to circumvent the problem. For the first one we aim to solve the density-weighted integrals analytically and for the second one we approximate the integrals numerically.

For the integrals in (9) and (10) to be tractable analytically, we need to assume that the posterior density $p(\mathbf{x}_n|\mathbf{z}_{1:n})$ is Gaussian for all time n which leads to a family of Kalman filter. On the other hand, when we resort to Monte Carlo approximation, we have a family of particle filters. A comparison between different methods in Bayesian tracking are summarized in Table 1.

2. A Family of Kalman Filters

The Kalman filter assumes that the posterior density $p(\mathbf{x}_n|\mathbf{z}_{1:n})$ at every time step n is Gaussian and, hence, parameterized by a mean $\hat{\mathbf{x}}_{n|n}$ and covariance $\mathbf{P}^{\mathbf{xx}}_{n|n}$ [1]. If $p(\mathbf{x}_{n-1}|\mathbf{z}_{1:n-1})$ is Gaussian, it can be proved that $p(\mathbf{x}_n|\mathbf{z}_{1:n})$ is also Gaussian, provided that certain assumptions hold [5]. First, $\mathbf{v}_{n-1}$ and $\mathbf{w}_n$ are drawn from Gaussian distributions of known parameters. Second, $\mathbf{f}_{n-1}(\mathbf{x}_{n-1}, \mathbf{v}_{n-1})$ is a known and linear function of $\mathbf{x}_{n-1}$ and $\mathbf{v}_{n-1}$. Third, $\mathbf{h}_n$ is a known linear function of $\mathbf{x}_n$ and $\mathbf{w}_n$.

With these assumptions, now we are able to derive an analytic form for the family of Kalman filters. Notice that most of the derivations are based on linear MMSE estimator which is exactly the original idea from Kalman [7]. However, in this paper, a bigger picture is given by the Bayesian framework thus we also need to interpret the Kalman filters from a Bayesian perspective. Therefore, in the following subsections, we not only show the original derivation from linear MMSE estimator, but also give a detailed derivation from a Bayesian perspective.

2.1. Derivations of Kalman Filter

2.1.1 A Perspective from Linear MMSE Estimator

Kalman [7] assumed that the updating estimate $\hat{\mathbf{x}}_{n|n}$ be linearly dependent on the current noisy measurement $\mathbf{z}_n$, that is,

$$\hat{\mathbf{x}}_{n|n} = \mathbf{A}\mathbf{z}_n + \mathbf{b} \tag{11}$$

where $\mathbf{A}$ and $\mathbf{b}$ are unknowns which are to be determined in a MMSE sense of random variable $\mathbf{x}_n$. Consider $\boldsymbol{\epsilon}^{\mathbf{x}}_n \triangleq \mathbf{x}_n - \hat{\mathbf{x}}_{n|n}$, by orthogonality principle, we need $\mathbf{A}$ and $\mathbf{b}$ such that

$$\mathbb{E}[\boldsymbol{\epsilon}^{\mathbf{x}}_n|\mathbf{z}_{1:n-1}] = \mathbf{0} \in \mathbb{R}^{n_x} \tag{12}$$

$$\mathbb{E}[\boldsymbol{\epsilon}^{\mathbf{x}}_n\mathbf{z}_n^T|\mathbf{z}_{1:n-1}] = \mathbf{0} \in \mathbb{R}^{n_x \times n_z} \tag{13}$$

where (12) is a condition for unbiased estimator and (13) is for uncorrelatedness between each component in estimation error and all components of the observation. Thus, for the unbiased condition, we have

$$\begin{aligned}
\mathbb{E}[\boldsymbol{\epsilon}^{\mathbf{x}}_n|\mathbf{z}_{1:n-1}] &= \mathbb{E}[(\mathbf{x}_n - \hat{\mathbf{x}}_{n|n})|\mathbf{z}_{1:n-1}] \\
&= \mathbb{E}[\mathbf{x}_n|\mathbf{z}_{1:n-1}] - \big(\mathbf{A}\mathbb{E}[\mathbf{z}_n|\mathbf{z}_{n-1}] + \mathbf{b}\big) \\
&= \hat{\mathbf{x}}_{n|n-1} - \mathbf{A}\hat{\mathbf{z}}_{n|n-1} - \mathbf{b} \\
&= \mathbf{0} \quad \Leftrightarrow \quad \mathbf{b} = \hat{\mathbf{x}}_{n|n-1} - \mathbf{A}\hat{\mathbf{z}}_{n|n-1}
\end{aligned} \tag{14}$$

and for the uncorrelatedness condition, we get

$$\begin{aligned}
\mathbb{E}[\boldsymbol{\epsilon}^{\mathbf{x}}_n\mathbf{z}_n^T|\mathbf{z}_{1:n-1}] &= \mathbb{E}[(\mathbf{x}_n - \hat{\mathbf{x}}_{n|n})\mathbf{z}_n^T|\mathbf{z}_{1:n-1}] \\
&= \mathbb{E}[(\mathbf{x}_n - \hat{\mathbf{x}}_{n|n})(\mathbf{z}_n - \hat{\mathbf{z}}_{n|n-1})^T|\mathbf{z}_{1:n-1}] \\
&= \mathbb{E}[(\mathbf{x}_n - \mathbf{A}\mathbf{z}_n - \mathbf{b})(\mathbf{z}_n - \hat{\mathbf{z}}_{n|n-1})^T|\mathbf{z}_{1:n-1}] \\
&= \mathbb{E}[\Big(\mathbf{x}_n - \hat{\mathbf{x}}_{n|n-1} - \mathbf{A}(\mathbf{z}_n - \hat{\mathbf{z}}_{n|n-1})\Big) \\
&\qquad \Big(\mathbf{z}_n - \hat{\mathbf{z}}_{n|n-1}\Big)^T|\mathbf{z}_{1:n-1}] \\
&= \mathbf{P}^{\mathbf{xz}}_{n|n-1} - \mathbf{A}\mathbf{P}^{\mathbf{zz}}_{n|n-1} \\
&= \mathbf{0} \quad \Leftrightarrow \quad \mathbf{A} = \mathbf{K}_n \triangleq \mathbf{P}^{\mathbf{xz}}_{n|n-1}\Big[\mathbf{P}^{\mathbf{zz}}_{n|n-1}\Big]^{-1}
\end{aligned} \tag{15}$$

Consequently, the updating estimate $\hat{\mathbf{x}}_{n|n}$ becomes

$$\begin{aligned}
\hat{\mathbf{x}}_{n|n} &= \mathbf{A}\mathbf{z}_n + \mathbf{b} \\
&= \mathbf{K}_n\mathbf{z}_n + \hat{\mathbf{x}}_{n|n-1} - \mathbf{K}_n\hat{\mathbf{z}}_{n|n-1} \\
&= \hat{\mathbf{x}}_{n|n-1} + \mathbf{K}_n\Big(\mathbf{z}_n - \hat{\mathbf{z}}_{n|n-1}\Big)
\end{aligned} \tag{16}$$

where $\mathbf{K}_n$ is the Kalman gain and $\boldsymbol{\epsilon}^{\mathbf{z}}_n \triangleq \mathbf{z}_n - \hat{\mathbf{z}}_{n|n-1}$ is the innovations (error in the observation prediction). On the other hand, the update covariance can be obtained from

$$\begin{aligned}
\mathbf{P}^{\mathbf{xx}}_{n|n} &= \mathbb{E}[\boldsymbol{\epsilon}^{\mathbf{x}}_n\boldsymbol{\epsilon}^{\mathbf{x}T}_n|\mathbf{z}_{1:n-1}] \\
&= \mathbb{E}[(\mathbf{x}_n - \hat{\mathbf{x}}_{n|n})(\mathbf{x}_n - \hat{\mathbf{x}}_{n|n})^T|\mathbf{z}_{1:n-1}] \\
&= \mathbb{E}[(\mathbf{x}_n - \hat{\mathbf{x}}_{n|n-1} - \mathbf{K}_n\Big(\mathbf{z}_n - \hat{\mathbf{z}}_{n|n-1}\Big)) \\
&\qquad (\mathbf{x}_n - \hat{\mathbf{x}}_{n|n-1} - \mathbf{K}_n\Big(\mathbf{z}_n - \hat{\mathbf{z}}_{n|n-1}\Big)^T|\mathbf{z}_{1:n-1}] \\
&= \mathbf{P}^{\mathbf{xx}}_{n|n-1} - \mathbf{K}_n\mathbf{P}^{\mathbf{zz}}_{n|n-1}\mathbf{K}_n^T
\end{aligned} \tag{17}$$

<table>
<tr><th colspan="7">Under the Bayesian framework</th></tr>
<tr><th>Family</th><th colspan="2">Method</th><th>Foundation</th><th>Perspective</th><th>Optimality</th><th>Complexity</th></tr>
<tr><td rowspan="7">KF</td><td colspan="2">Linear KF</td><td rowspan="6">Gaussian assumption</td><td rowspan="7">State-space</td><td>Optimal</td><td>Low</td></tr>
<tr><td rowspan="5">Sigma point class</td><td>Extended KF</td><td rowspan="9">Suboptimal</td><td rowspan="2">High</td></tr>
<tr><td>Finite difference KF</td></tr>
<tr><td>Unscented KF</td><td rowspan="2">Low</td></tr>
<tr><td>Spherical simplex KF</td></tr>
<tr><td>Gauss-Hermite KF</td><td rowspan="5">High</td></tr>
<tr><td colspan="2">Monte Carlo KF</td><td>Gaussian assumption and LLN</td></tr>
<tr><td rowspan="3">PF</td><td rowspan="3">Sequential importance sampling</td><td>Bootstrap PF</td><td rowspan="3">LLN</td><td rowspan="3">Transition and likelihood densities</td></tr>
<tr><td>Optimal PF</td></tr>
<tr><td>Auxiliary PF</td></tr>
</table>

Table 1: Comparison between different methods in Bayesian tracking.

2.1.2 A Perspective from Bayes' Theorem

Consider $\mathbf{q}_n \triangleq \begin{bmatrix}\mathbf{x}_n \\ \mathbf{z}_n\end{bmatrix}$, by Bayes' theorem, we have

$$p(\mathbf{x}_n|\mathbf{z}_{1:n}) = \frac{p(\mathbf{q}_n|\mathbf{z}_{1:n-1})}{p(\mathbf{z}_n|\mathbf{z}_{1:n-1})} \tag{18}$$

Let all densities in (18) be Gaussian so that

$$\begin{aligned} p(\mathbf{x}_n|\mathbf{z}_{1:n}) &= \mathcal{N}\left(\mathbf{x}_n, \hat{\mathbf{x}}_{n|n}, \mathbf{P}^{\mathbf{xx}}_{n|n}\right) \propto e^{-\frac{1}{2}a} \\ p(\mathbf{q}_n|\mathbf{z}_{1:n-1}) &= \mathcal{N}\left(\mathbf{q}_n, \hat{\mathbf{q}}_{n|n-1}, \mathbf{P}^{\mathbf{qq}}_{n|n-1}\right) \propto e^{-\frac{1}{2}b} \\ p(\mathbf{z}_n|\mathbf{z}_{1:n-1}) &= \mathcal{N}\left(\mathbf{z}_n, \hat{\mathbf{z}}_{n|n-1}, \mathbf{P}^{\mathbf{zz}}_{n|n-1}\right) \end{aligned} \tag{19}$$

According to (18), the random vector $\mathbf{x}_n$ in $p(\mathbf{x}_n|\mathbf{z}_{1:n})$ is only related to $p(\mathbf{q}_n|\mathbf{z}_{1:n-1})$. Therefore, the relation between the statistics in (19) can be found by equalizing terms of $\mathbf{x}_n$ in a and b given by

$$\begin{aligned} a &= \left(\mathbf{x}_n - \hat{\mathbf{x}}_{n|n}\right)^T \left[\mathbf{P}^{\mathbf{xx}}_{n|n}\right]^{-1} \left(\mathbf{x}_n - \hat{\mathbf{x}}_{n|n}\right) \\ &= \mathbf{x}_n^T \left[\mathbf{P}^{\mathbf{xx}}_{n|n}\right]^{-1} \mathbf{x}_n - \mathbf{x}_n^T \left[\mathbf{P}^{\mathbf{xx}}_{n|n}\right]^{-1} \hat{\mathbf{x}}_{n|n} - \cdots \end{aligned} \tag{20}$$

$$\begin{aligned} b &= \left(\mathbf{q}_n - \hat{\mathbf{q}}_{n|n-1}\right)^T \left[\mathbf{P}^{\mathbf{qq}}_{n|n}\right]^{-1} \left(\mathbf{q}_n - \hat{\mathbf{q}}_{n|n-1}\right) \\ &= \begin{bmatrix}\mathbf{x}_n - \hat{\mathbf{x}}_{n|n-1} \\ \mathbf{z}_n - \hat{\mathbf{z}}_{n|n-1}\end{bmatrix}^T \begin{bmatrix}\mathbf{P}^{xx}_{n|n-1} & \mathbf{P}^{xz}_{n|n-1} \\ \mathbf{P}^{zx}_{n|n-1} & \mathbf{P}^{zz}_{n|n-1}\end{bmatrix}^{-1} \begin{bmatrix}\mathbf{x}_n - \hat{\mathbf{x}}_{n|n-1} \\ \mathbf{z}_n - \hat{\mathbf{z}}_{n|n-1}\end{bmatrix} \end{aligned} \tag{21}$$

To proceed, we define the inverse of the block matrix as

$$\begin{bmatrix}\mathbf{C}_{11} & \mathbf{C}_{12} \\ \mathbf{C}_{21} & \mathbf{C}_{22}\end{bmatrix} \triangleq \begin{bmatrix}\mathbf{P}^{xx}_{n|n-1} & \mathbf{P}^{xz}_{n|n-1} \\ \mathbf{P}^{zx}_{n|n-1} & \mathbf{P}^{zz}_{n|n-1}\end{bmatrix}^{-1} \tag{22}$$

and use the Woodbury matrix identity, we obtain

$$\begin{aligned} \mathbf{C}_{11} &= \left[\mathbf{P}^{\mathbf{xx}}_{n|n-1} - \mathbf{P}^{\mathbf{xz}}_{n|n-1}\left(\mathbf{P}^{\mathbf{zz}}_{n|n-1}\right)^{-1}\mathbf{P}^{\mathbf{zx}}_{n|n-1}\right]^{-1} \\ \mathbf{C}_{12} &= -\mathbf{C}_{11}\mathbf{P}^{\mathbf{xz}}_{n|n-1}\left(\mathbf{P}^{\mathbf{zz}}_{n|n-1}\right)^{-1} \\ \mathbf{C}_{21} &= -\mathbf{C}_{22}\mathbf{P}^{\mathbf{zx}}_{n|n-1}\left(\mathbf{P}^{\mathbf{xx}}_{n|n-1}\right)^{-1} \\ \mathbf{C}_{22} &= \left[\mathbf{P}^{\mathbf{zz}}_{n|n-1} - \mathbf{P}^{\mathbf{zx}}_{n|n-1}\left(\mathbf{P}^{\mathbf{xx}}_{n|n-1}\right)^{-1}\mathbf{P}^{\mathbf{xz}}_{n|n-1}\right]^{-1} \end{aligned} \tag{23}$$

Now, use (23) to expand (21), we obtain

$$b = \mathbf{x}_n^T\mathbf{C}_{11}\mathbf{x}_n + \mathbf{x}_n^T\left[-\mathbf{C}_{11}\hat{\mathbf{x}}_{n|n-1} + \mathbf{C}_{12}\left(\mathbf{z}_n - \hat{\mathbf{z}}_{n|n-1}\right)\right] + \cdots \tag{24}$$

Comparing the terms in (20) with the terms in (24), it follows immediately that $\mathbf{P}^{\mathbf{xx}}_{n|n} = \mathbf{C}_{11}^{-1}$ and

$$\begin{aligned} \hat{\mathbf{x}}_{n|n} &= \mathbf{P}^{\mathbf{xx}}_{n|n}\mathbf{C}_{11}\hat{\mathbf{x}}_{n|n-1} - \mathbf{P}^{\mathbf{xx}}_{n|n}\mathbf{C}_{12}\left(\mathbf{z}_n - \hat{\mathbf{z}}_{n|n-1}\right) \\ &= \hat{\mathbf{x}}_{n|n-1} + \underbrace{\mathbf{P}^{\mathbf{xz}}_{n|n-1}\left(\mathbf{P}^{\mathbf{zz}}_{n|n-1}\right)^{-1}}_{\mathbf{K}_n}\left(\mathbf{z}_n - \hat{\mathbf{z}}_{n|n-1}\right) \end{aligned} \tag{25}$$

$$\begin{aligned} \mathbf{P}^{\mathbf{xx}}_{n|n} &= \mathbf{P}^{\mathbf{xx}}_{n|n-1} - \mathbf{P}^{\mathbf{xz}}_{n|n-1}\left(\mathbf{P}^{\mathbf{zz}}_{n|n-1}\right)^{-1}\mathbf{P}^{\mathbf{zx}}_{n|n-1} \\ &= \mathbf{P}^{\mathbf{xx}}_{n|n-1} - \mathbf{K}_n\mathbf{P}^{\mathbf{zz}}_{n|n-1}\mathbf{K}_n^T \end{aligned} \tag{26}$$

which is the same result that we have derived from linear MMSE estimator.

2.2. Linear Kalman Filter (LKF)

For the system dynamics to be linear, the transformation can be described by

$$\begin{aligned} \mathbf{f}_{n-1}(\mathbf{x}_{n-1}) &= \mathbf{F}_{n-1}\mathbf{x}_{n-1} \\ \mathbf{h}_n(\mathbf{x}_n) &= \mathbf{H}_n\mathbf{x}_n \end{aligned} \tag{27}$$

where $\mathbf{F}_n \in \mathbb{R}^{n_x \times n_x}$, $\mathbf{H}_n \in \mathbb{R}^{n_z \times n_x}$ and $\mathbf{x}_n \in \mathbb{R}^{n_x}$. In this case, we call this type of Kalman filter as linear Kalman filter and the resulting estimate $\hat{\mathbf{x}}_{n|n}$ is optimal in a minimum mean squared error sense. The procedure for linear Kalman filter is given by Algorithm 1. Notice that all posterior statistics are computed in a recursive way.

Algorithm 1: Linear Kalman Filter

initialize $\hat{\mathbf{x}}_0$ and $\mathbf{P}_0^{\mathbf{xx}}$;
repeat
$\hat{\mathbf{x}}_{n|n-1} \leftarrow \mathbf{F}_{n-1}\hat{\mathbf{x}}_{n-1|n-1}$;
$\mathbf{P}^{\mathbf{xx}}_{n|n-1} \leftarrow \mathbf{F}_{n-1}\mathbf{P}^{\mathbf{xx}}_{n-1|n-1}\mathbf{F}^T_{n-1} + \mathbf{Q}_n$;
$\hat{\mathbf{z}}_{n|n-1} \leftarrow \mathbf{H}_n\hat{\mathbf{x}}_{n|n-1}$;
$\mathbf{P}^{\mathbf{zz}}_{n|n-1} \leftarrow \mathbf{H}_n\mathbf{P}^{\mathbf{xx}}_{n|n-1}\mathbf{H}^T_n + \mathbf{R}_n$;
$\mathbf{P}^{\mathbf{xz}}_{n|n-1} \leftarrow \mathbf{P}^{\mathbf{xx}}_{n|n-1}\mathbf{H}^T_n$;
$\mathbf{K}_n \leftarrow \mathbf{P}^{\mathbf{xz}}_{n|n-1}\left(\mathbf{P}^{\mathbf{zz}}_{n|n-1}\right)^{-1}$;
$\hat{\mathbf{x}}_{n|n} \leftarrow \hat{\mathbf{x}}_{n|n-1} + \mathbf{K}_n\left(\mathbf{z}_n - \hat{\mathbf{z}}_{n|n-1}\right)$;
$\mathbf{P}^{\mathbf{xx}}_{n|n} \leftarrow \mathbf{P}^{\mathbf{xx}}_{n|n-1} - \mathbf{K}_n\mathbf{P}^{\mathbf{zz}}_{n|n-1}\mathbf{K}^T_n$;

2.3. Nonlinear Trackers

Notice that in the nonlinear case, part of the Gaussian assumption is violated, so the posterior density $p(\mathbf{x}_n|\mathbf{z}_{1:n})$ at every time step n cannot be guaranteed to be Gaussian, they may be non-Gaussian. However, we still need the Gaussianity so that we are able to evaluate the density-weighted integrals. Hence, for those system dynamics to be nonlinear, e.g., the system cannot be expressed in linear transformation given by (27), approximation techniques and linearization of the system dynamics $\mathbf{f}_{n-1}$ and $\mathbf{h}_n$ are needed. For example, Taylor expansion is used to approximate the nonlinear function in extended Kalman filter (EKF) [6] and sigma points are generated to evaluate the Gaussian-weighted integrals in sigma point class kalman filters [4].

We will not use the nonlinear family of Kalman filters in our experiments due to the difficulty of modeling the system dynamics. More details about this can be found in [6] and [4].

3. A Family of Particle Filters

Without Gaussian assumption, the evaluation of density-weighted integrals become intractable and we do not have a closed-form solution. Thus, Monte Carlo methods come to address the problem of integral evaluation. This type of Bayesian solution is then known as a family of particle filters. Therefore, in the following subsection, we firstly introduce general concepts of importance sampling, then follow by applying sequential importance sampling (SIS) to form the basis of particle filters.

3.1. General Concepts of Importance Sampling

Suppose we need to evaluate the integrals of the form

$$\hat{\mathbf{f}}(\mathbf{x}) = \int \mathbf{f}(\mathbf{x})p(\mathbf{x})d\mathbf{x} \tag{28}$$

For those solutions do not have analytic forms, we can approximate the integral by generating a set of samples $\{\mathbf{x}^{(i)}, i = 1, 2, ..., N_s\}$ from the density $p(\mathbf{x})$. Hence, the Monte Carlo integration is then given by

$$\begin{aligned} \hat{\mathbf{f}}(\mathbf{x}) &= \int \mathbf{f}(\mathbf{x}) \sum_{i=1}^{N_s} w_i \delta\left(\mathbf{x} - \mathbf{x}^{(i)}\right) d\mathbf{x} \\ &= \sum_{i=1}^{N_s} w_i \mathbf{f}\left(\mathbf{x}^{(i)}\right) \end{aligned} \tag{29}$$

where $w_i = \frac{1}{N}, \forall i$. Here, w_i represents the importance of $\mathbf{x}^{(i)}$ which are all equally important to the summation due to directly sampling from a known density $p(\mathbf{x})$. However, when the density $p(\mathbf{x})$ is difficult to sample, a well known and easily sampled probability density function $q(\mathbf{x})$ whose support is greater than or equal to the support of $p(\mathbf{x})$ can come to play. The density $q(\mathbf{x})$ is known as *importance density* and the weighting factor can be computed by

$$w(\mathbf{x}) = \frac{p(\mathbf{x})}{q(\mathbf{x})} \tag{30}$$

3.2. SIS Particle Filter

The posterior density is given by

$$p(\mathbf{x}|\mathbf{z}_{1:n}) = \sum_{i=1}^{N_s} w_n^{(i)} \delta\left(\mathbf{x}_n - \mathbf{x}_n^{(i)}\right) \tag{31}$$

where the SIS recursive weight update equation is given by

$$\tilde{w}_n^{(i)} = w_{n-1}^{(i)} \frac{p\left(\mathbf{z}_n|\mathbf{x}_n^{(i)}\right) p\left(\mathbf{x}_n^{(i)}|\mathbf{x}_{n-1}^{(i)}\right)}{q\left(\mathbf{x}_n^{(i)}|\mathbf{x}_{n-1}^{(i)}, \mathbf{z}_n\right)}, \forall i \tag{32}$$

and the weights should be normalized by

$$w_n^{(i)} = \frac{\tilde{w}_n^{(i)}}{\sum_{i=1}^{N_s} \tilde{w}_n^{(i)}}, \forall i \tag{33}$$

The derivation of sequential importance sampling particle filters can be found in the literature [4]. Notice that there are several significant practical problems in (32) since we need the likelihood function $p(\mathbf{z}_n|\mathbf{x}_n)$, state transition function $p(\mathbf{x}_n|\mathbf{x}_{n-1})$ and importance density $q(\mathbf{x}_n|\mathbf{x}_{n-1},\mathbf{z}_n)$ which are hard to find.

Algorithm 2: SIS Particle Filter

initialize $\mathbf{x}_0^{(i)} \sim q(\mathbf{x}_0), i = 1, 2, ..., N_s$ and $w_0^{(i)} \leftarrow \frac{1}{N_s}$;
repeat
$\mathbf{x}_n^{(i)} \sim q\,(\,\mathbf{x}_n|\mathbf{x}_{n-1}^{(i)}, \mathbf{z}_n), \forall i;$
$\tilde{w}_n^{(i)} \leftarrow w_{n-1}^{(i)} \frac{p\left(\mathbf{z}_n|\mathbf{x}_n^{(i)}\right)p\left(\mathbf{x}_n^{(i)}|\mathbf{x}_{n-1}^{(i)}\right)}{q\left(\mathbf{x}_n^{(i)}|\mathbf{x}_{n-1}^{(i)},\mathbf{z}_n\right)}, \forall i;$
$w_n^{(i)} \leftarrow \frac{\tilde{w}_n^{(i)}}{\sum_{i=1}^{N_s}\tilde{w}_n^{(i)}}, \forall i;$
$\mathbf{x}_{n-1}^{(i)} \leftarrow \mathbf{x}_n^{(i)}, \forall i;$
$w_{n-1}^{(i)} \leftarrow w_n^{(i)}, \forall i;$
$\hat{\mathbf{x}}_n \leftarrow \sum_{i=1}^{N_s} w_n^{(i)}\mathbf{x}_n^{(i)};$
$\mathbf{P}_n^{\mathbf{xx}} \leftarrow \sum_{i=1}^{N_s} w_n^{(i)}\left(\mathbf{x}_n^{(i)} - \hat{\mathbf{x}}_n\right)\left(\mathbf{x}_n^{(i)} - \hat{\mathbf{x}}_n\right)^T$

3.2.1 Bootstrap Particle Filter (BPF)

One of the easiest to implement, and thus one of the most widely used, SIS particle filters is the bootstrap particle filter. The transition density is chosen as the importance density, that is

$$q(\mathbf{x}_n|\mathbf{x}_{n-1},\mathbf{z}_n) = p(\mathbf{x}_n|\mathbf{x}_{n-1}) \tag{34}$$

Hence, the weight update equation in (32) becomes

$$\tilde{w}_n^{(i)} = w_{n-1}^{(i)} p\left(\mathbf{z}_n|\mathbf{x}_n^{(i)}\right) \tag{35}$$

To avoid degeneracy, resampling is required if

$$N_{eff} \ll N_s \tag{36}$$

where

$$N_{eff} = \frac{1}{\sum_{i=1}^{N_s}\left(w_n^{(i)}\right)} \tag{37}$$

4. Observation Model

For the family of Kalman filters, the observation model is required to get the observation or measurement $\mathbf{z}_n$ at time n. Basically, we obtain $\mathbf{z}_n$ by detection and even incorporate some prior knowledge implicitly.

For the bootstrap particle filters, the observation model is to obtain the likelihood function $p(\mathbf{z}_n|\mathbf{x}_n)$.

Hence, in the following subsection, we present two different observation models including histogram color model and optical flow. Notice that optical flow in nature have some prior knowledge implicitly because we aim to see will there be any boost in performance if we incorporate prior knowledge explicitly by applying Bayesian methods.

4.1. Histogram Color Model

Suppose that $\mathbf{q}^* \in \mathbb{R}^M$ is the reference histogram color model and $\mathcal{D}_n$ is the current frame data, then a simple observation model for the family of Kalman filter is given by

$$\mathbf{z}_n = \arg\min_{\mathbf{z}} D\left[\mathbf{q}^*, \mathbf{q}_n(\mathbf{z}, \mathcal{D}_n)\right] \tag{38}$$

From [10], the observation model for the bootstrap particle filters is given by

$$p(\mathbf{z}_n|\mathbf{x}_n) \propto e^{-\lambda D^2\left[\mathbf{q}^*, \mathbf{q}_n(\mathbf{x_n}, \mathcal{D}_n)\right]} \tag{39}$$

where D is derived from the Bhattacharyya similarity coefficient, and defined as

$$D\left[\mathbf{q}^*, \mathbf{q}_n(\mathbf{z}, \mathcal{D}_n)\right] = \left[1 - \sum_{i=1}^{M}\sqrt{q_i^*\left(q_n(\mathbf{z}, \mathcal{D}_n)\right)_i}\right]^{\frac{1}{2}} \tag{40}$$

4.2. Optical Flow

Optical flow is defined as the change of structured light in the image. It consists of vectors and each of them is assigned to certain pixel position, that points to where that pixel can be found in next image frame. Two key assumptions are as follows

- Brightness constancy: the appearance of the image patches do not change.

$$E(x, y, t) = E(x + dx, y + dy, t + dt) \tag{41}$$

- Small motion: points do not move very far.

$$E_x\frac{dx}{dt} + E_y\frac{dy}{dt} + E_t = 0 \tag{42}$$

5. Dataset

5.1. Multiple Object Tracking Benchmark

In order to demonstrate the power of incorporating prior knowledge from a Bayesian point of view. Particularly, we pick a challenging problem, that is, multiple object tracking which we may have around a hundred of objects in a scene. In this project, we use Multiple Object Tracking Benchmark [9] as our dataset to conduct the simulations and experiments.

The dataset [9] consists of tens of video represented by image sequences and their corresponding annotations are bounding boxes for each target in the video. Our target class can be divided into three groups including moving or standing pedestrians, people that are not in an upright position or artificial representations of humans, vehicles and occluders. Fig. 1 is an illustration of the dataset and our target of interests.

5.2. Evaluation Metrics

The Multiple Object Tracking Precision [9] is the average similarity between all true positives and their corresponding ground truth targets. For bounding box overlap, this is computed as

$$\text{MOTP} = \frac{\sum_{t,i} d_{t,i}}{\sum_t c_t} \tag{43}$$

where c_t denotes the number of matches in frame t and $d_{t,i}$ is the bounding box overlap of target i with its assigned ground truth object.

Each ground truth (GT) trajectory can be classified as mostly tracked (MT), partially tracked (PT), and mostly lost (ML). This is done based on how much of the trajectory is recovered by the tracking algorithm. A target is mostly tracked if it is successfully tracked for at least 80% of its life span [9].

6. Experiments

6.1. Settings

We use two observation models as our baselines including histogram color model [10] (HCM) and optical flow [2] (OF). For our proposed recursive Bayesian tracking methods, it consists of prior knowledge and observation. Two assumptions are tested, e.g., Gaussian assumption and Monte Carlo methods. In the case of Gaussian assumption, we have LKF+HCM and LKF+OF. For the case of applying Monte Carlo method, we use BPF+HCM.

The tracking results we show in the following subsection use the video MOT17-02 in MOT dataset [9]. It has 30 fps, 1920×1080 pixel resolution, 600 image sequences ($n_{max} = 600$) and 62 targets in total.

6.2. Tracking Results

We particularly pick target 3 to visualize the tracking behaviors among different methods. In the early stage of the video, say $n = 6$, every method has similar behavior and there is no such difference between them (Fig. 2). As time goes by, when $n = 169$, the position and scale of target 3 have changed dramatically (Fig. 3). In such situation, LKF+OF has much more accurate estimates than the others. Notice that baseline-2 outperforms baseline-1 and this can be explained by the implicit prior in optical flow. Fig. 4 shows the situation when occlusion occurs on our target. As we can expect, HCM cannot handle occlusion, thus it performs poorest.

Figure 2: The scene is still very similar to the first frame ($n = 6$, target 3).

Figure 3: The changes of position and scale ($n = 169$, target 3).

Fig. 5 and 6 show the trajectories of (x_n, y_n) and (w_n, h_n) using different tracking methods on target 3. Finally, we evaluate and summarize the overall performance in terms of MT, PT, ML and MOTP for every method in Table 2. To sum up, methods with prior knowledge, i.e., LKF+HCM, BPF+HCM and LKF+OF have much more accurate estimates than baseline-1 (HCM) and baseline-2 (OF).

Method	GT	MT	PT	ML	MOTP
HCM (basesline-1)	62	7	39	16	65.1
LKF+HCM	62	24	32	6	71.7
BPF+HCM	62	11	43	8	67.7
OF (basesline-2)	62	5	12	45	68.0
LKF+OF	62	21	33	8	69.9

Table 2: Quantitative results of different tracking methods.

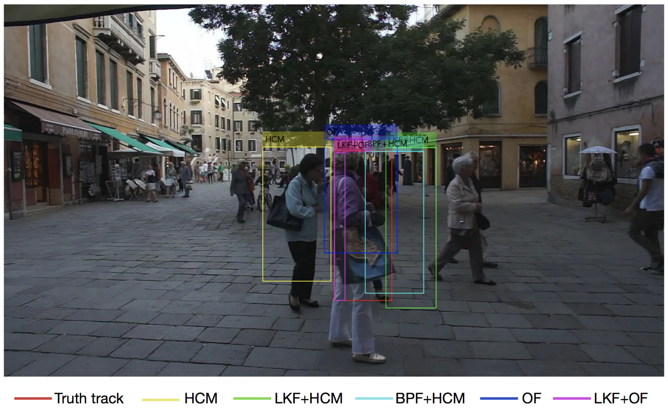

Figure 4: Occlusion ($n = 257$, target 3).

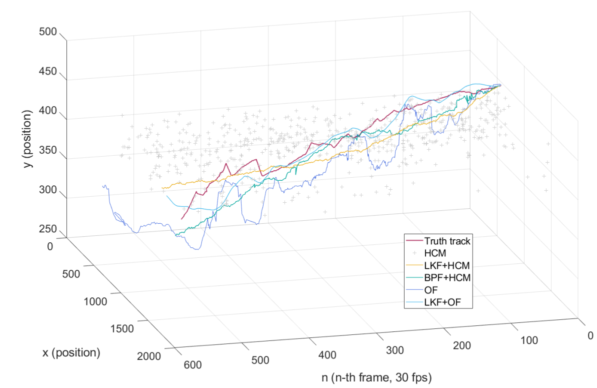

Figure 5: A visualization of (x_n, y_n) using different tracking methods (target 3).

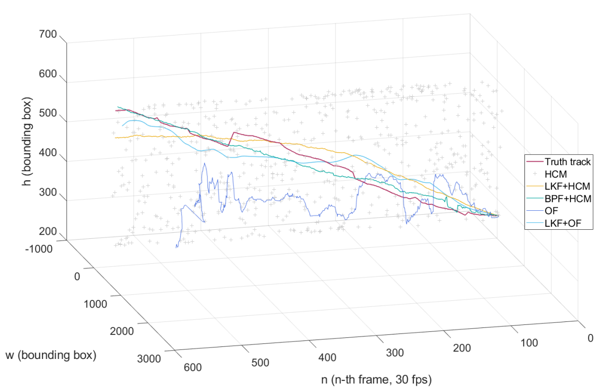

Figure 6: A visualization of (w_n, h_n) using different tracking methods (target 3).

7. Conclusion

In recursive Bayesian tracking, assumptions are made to circumvent the difficulties. Kalman filters and particle filters are special cases in the Bayesian framework. In the Bayesian view point, the LMMSE estimator can be actually explained by conditional density propagation. Finally, all parameters and statistics are updated in a recursive manner which is computationally efficient.

We have demonstrated the power of recursive Bayesian tracking in a real-world multiple object tracking problem on a standard dataset. Notice that parameter learning techniques are required to formulate the problem to the Bayesian framework and those are hard. However, with correct parameters, estimates are much more accurate by incorporating correct prior according to the experimental results. The model for BPF+HCM may be incorrect (the underlying likelihood function may be Gaussian) which indicates that the function family of the model is very difficult to find and plays a crucial part in the Bayesian framework.

References

[1] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on signal processing*, 50(2):174–188, 2002.

[2] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. *International journal of computer vision*, 12(1):43–77, 1994.

[3] Z. Ghahramani and G. E. Hinton. Parameter estimation for linear dynamical systems. Technical report, Technical Report CRG-TR-96-2, University of Totronto, Dept. of Computer Science, 1996.

[4] A. J. Haug. *Bayesian estimation and tracking: a practical guide*. John Wiley & Sons, 2012.

[5] Y. Ho and R. Lee. A bayesian approach to problems in stochastic estimation and control. *IEEE Transactions on Automatic Control*, 9(4):333–339, 1964.

[6] S. J. Julier and J. K. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, Mar 2004.

[7] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

[8] L. Lennart. System identification theory for the user, 1999.

[9] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler. MOT16: A benchmark for multi-object tracking. *arXiv:1603.00831 [cs]*, Mar. 2016. arXiv: 1603.00831.

[10] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. Color-based probabilistic tracking. In *European Conference on Computer Vision*, pages 661–675. Springer, 2002.