VISUAL TRACKING USING HIGH-ORDER MONTE CARLO MARKOV CHAIN

Pan Pan and Dan Schonfeld

ECE Department, University of Illinois at Chicago
851 S. Morgan St., Chicago, IL, USA 60607
Email: {ppan3, dans}@uic.edu

ABSTRACT

In this paper, we discard the first-order Markov state-space model commonly used in visual tracking and present a framework of visual tracking using high-order Monte Carlo Markov chain. By using graphic models to obtain the conditional independence properties, we derive the general expression of posterior density function for the mth-order hidden Markov model. We subsequently use Sequential Importance Sampling method to estimate the posterior density and obtain the high-order particle filtering algorithm for tracking. Experimental results show the superior performance of our proposed algorithm to traditional first-order particle filtering tracking algorithm, i.e. particle filtering derived based on first-order Markov chain.

Index Terms— High-order Markov chain, graphic models, particle filtering, visual tracking

1. INTRODUCTION

Many real world applications require robust and accurate object tracking, including video surveillance, human computer interfaces, robot localization, etc. [1][2]. Given the state $x_{0:k}$ from time 0 up to $k$ and the observations $z_{1:k}$, visual tracking is usually interpreted as to find the posterior density function $p(x_{0:k}|z_{1:k})$, where $x_{0:k}$ is hidden and can only be estimated through the observations $z_{1:k}$.

Sequential Importance Sampling (SIS) is widely used as a density estimation method, e.g. estimation of the posterior density function. The concomitant particle filters, known also as Sequential Monte Carlo filters, have become very popular for video tracking. In particle filtering, the random samples, called particles, are generated from a proposal density and are used to evaluate the importance weights, which are normalized and subsequently used to the estimate the posterior density function.

These days, graphic models [3] have been used as a tool for applications in pattern recognition and machine learning. It provides a more simple way to visualize the structure of the probability model. Moreover, the conditional dependence properties can be obtained by inspection of the graph. For example, we can obtain the conditional independence properties by applying the Markov properties to the moral graph of a directed graph [3].

So far, the common model adopted in visual tracking is a first-order hidden Markov model (HMM), where $x_k$ is only depends on the previous state $x_{k-1}$ [1][4]. Although the assumption of the first-order Markov model can simplify the expression of the posterior density and the implementation of visual tracking, it has the following disadvantages. First, the first-order model is not generally true and cannot accurately characterize the dynamics of moving objects. For example, the particle filtering algorithm based on the first-order model cannot work efficiently on a moving object with dynamics of order three. Secondly, if the particles at time $k-1$ are lost or delayed, the propagation for particles cannot be maintained. Therefore, we hope to extend the first-order HMM model to higher order and achieve more robust tracking.

In this paper, we discard the first-order Markov state-space model commonly used in visual tracking and present a framework of visual tracking using mth-order Monte Carlo Markov chain in Section 2. In Section 3, we use Sequential Importance Sampling technique to estimate the posterior density function and get the high-order particle filtering tracking algorithm. Simulation results compared with first-order particle filtering are given in Section 4, followed by conclusion in Section 5.

2. HIGH-ORDER MONTE CARLO MARKOV CHAIN MODEL

2.1. Mth-Order Markov Chain Model

Compared with the traditional first-order particle filtering derived based on the first-order Markov chain, the high-order Markov chain in this paper stands for a mth-order Markov chain when $m \geq 2$. If the state-space model is a mth-order Markov chain, the current state $x_k$ depends on the past $m$ states, i.e. $x_{k-1}, x_{k-2}, \ldots, x_{k-m}$. Therefore, we have

$$p(x_k|x_{k-1}, x_{k-2}, \ldots, x_0) = p(x_k|x_{k-1}, x_{k-2}, \ldots, x_{k-m}).$$

Figure 1 gives an example of a second-order hidden Markov model, i.e. $m = 2$. In Fig. 1, the circle nodes represent the states of the object, e.g. $x_3$. The square nodes
denote the observations associated with the hidden states, e.g. $z_3$. Solid lines represent the first-order hidden Markov model and the dashed lines denote the higher-order dependencies.

### 2.2. Graphical Model of Mth-Order Hidden Markov Model

The mth-order HMM is an acyclic directed graph [3]. The moral graph associated with a directed graph is the undirected graph on the same vertex set and with an edge set obtained by including all edges in the directed graph together with all edges necessary to eliminate forbidden Wermuth configurations [3]. For example, the moral graph of the state-space model presented in Fig. 1 is given by Fig. 2.

We use $a \perp b|c$ to denote random variables $a$, $b$ are independent conditional on $c$. By utilizing the Markov properties [3] on the moral graph of the mth-order hidden Markov model, we can obtain the following conditional independence:

$$
\begin{align*}
z_k &\perp z_{1:k-1}|x_{0:k}, 
\end{align*}
$$

Therefore, we have

$$
\begin{align*}
p(z_k|x_{0:k}, z_{1:k-1}) &= p(z_k|x_k); \\
p(x_k|x_{0:k-1}, z_{1:k-1}) &= p(x_k|x_{m:k-1}).
\end{align*}
$$

### 2.3. Posterior Density

The conditional independence properties obtained in Section 2.2 are to be used to derive the expression of the posterior density. Parallel with the derivation under first-order HMM in [4], we formulate the posterior density function $p(x_{0:k}|z_{1:k})$ as:

$$
\begin{align*}
p(x_{0:k}|z_{1:k}) &= \frac{p(z_k, x_{0:k}, z_{1:k-1})}{p(z_k, z_{1:k-1})} \\
&= \frac{p(z_k|x_{0:k}, z_{1:k-1})p(x_{0:k}|z_{1:k-1})}{p(z_k, z_{1:k-1})} \\
&= \frac{p(z_k|x_{0:k}, z_{1:k-1})p(x_{0:k}|z_{1:k-1})}{p(z_k|x_{1:k})} \\
&= \frac{p(z_k|x_{0:k}, z_{1:k-1})p(x_{0:k-1}|z_{1:k-1})p(x_{0:k-1}|z_{1:k-1})}{p(z_k|x_{1:k})} \\
&= \frac{p(z_k|x_{0:k}, z_{1:k-1})p(x_{0:k-1}|z_{1:k-1})p(x_{0:k-1}|z_{1:k-1})}{p(z_k|x_{1:k})} \\
&\propto p(z_k|x_{0:k}, z_{1:k-1})p(x_{0:k}|z_{1:k-1})p(x_{0:k-1}|z_{1:k-1})p(x_{0:k-1}|z_{1:k-1}).
\end{align*}
$$

In (3) (4) (5) and (7), we use the conditional probability rule $p(a|b) = p(a, b)/p(b)$ and $p(a, b|c) = p(a|b, c)p(b|c)$, given $a$, $b$, $c$ are random variables. (1) and (2) are used to simplify (8) to (9). When $m = 1$, i.e. the state-space model is a first-order HMM, (9) reduces to the expression presented in [4].

### 3. HIGH-ORDER PARTICLE FILTERING

Now our object is to estimate the posterior function derived in Section 2. Different density estimation methods can be applied to our framework, e.g. Gaussian Mixture model, Kernel density estimation, etc. In this paper, we exploit the Sequential Importance Sampling [4] method.

#### 3.1. Weight Updating

We denote $\{x^{i}_{0:k}, w^{i}_{k}\}_{i=1}^{N}$ as a random measure that characterizes the posterior pdf $p(x_{0:k}|z_{1:k})$, where $x_{0:k} = \{x_j, j = 0, \ldots, k\}$ is the set of all states up to time $k$; $\{x^{i}_{0:k}, i = 0, \ldots, N\}$ is a set of particles with associated weights $\{w^{i}_{k}, i = 0, \ldots, N\}$. The weights are normalized such that $\sum_{i=1}^{N} w^{i}_{k} = 1$. Then the posterior density can be approximated as [4]

$$
\begin{align*}
p(x_{0:k}|z_{1:k}) &\approx \sum_{i=1}^{N} w^{i}_{k}\delta(x_{0:k} - x^{i}_{0:k}).
\end{align*}
$$

According to the importance sampling theory, if the particles $\{x^{i}_{0:k}\}$ are drawn from an importance density
is good, represented as having higher observation likelihood 

will be assigned higher weight if (i) the final destination 

\( q(\mathbf{x}_k|z_{1:k}) \) is the normalized weight, given by [4]

\[
w_k^i \propto \frac{p(x_k^i|z_{1:k})}{q(x_k^i|z_{1:k})}.
\]  

(11)

The importance density is chosen to factorize such that

\[
q(\mathbf{x}_0:0|z_{1:k}) = q(\mathbf{x}_k|\mathbf{x}_0:0:k-1, z_{1:k})q(\mathbf{x}_0:k-1|z_{1:k-1})
\]

\[
= q(\mathbf{x}_k|\mathbf{x}_{k-m:k-1}, \mathbf{z}_k)q(\mathbf{x}_0:k-1|z_{1:k-1}).
\]

(12)

Submitting (9) and (12) into (11), the weight update equation can be obtained

\[
w_k^i \propto \frac{p(x_k^i|z_{1:k})}{q(x_k^i|z_{1:k})} \frac{p(x_k^i|z_{k-m:k-1})}{q(x_k^i|z_{k-m:k-1})} \frac{p(x_0^i|z_{1:k-1})}{q(x_0^i|z_{1:k-1})} = w_k^{i-1} \frac{p(z_k^i|x_k^i)}{q(x_k^i|z_{k-m:k-1}, \mathbf{z}_k)},
\]  

(13)

where \( p(z_k^i|x_k^i) \) is the likelihood, and \( p(x_k^i|z_{k-m:k-1}) \) is the transition probability, and \( q(x_k^i|z_{k-m:k-1}, \mathbf{z}_k) \) is the proposal density. Therefore, the posterior filtered density \( p(\mathbf{x}_k|z_{1:k}) \) can be approximated as

\[
p(\mathbf{x}_k|z_{1:k}) \approx \sum_{i=1}^{N} w_k^i \delta(x_k - x_k^i).
\]  

(14)

3.2. Discussion and Analysis

We now compare particle filtering under first-order HMM model and mth-order HMM model. A summary is provided in Table 1.

In mth-order particle filtering, the tracking state is a matrix \( \mathbf{x}_{k-m+1:k} \), which is a combined description of the object over \( m \) frames. And the particles \( \{\mathbf{x}_{k-m+1:k}^i\}_{i=1}^{N} \) from frame \( k-m+1 \) to \( k \) need to be stored. We could consider the particles \( \{\mathbf{x}_{k-m+1:k}^i\}_{i=1}^{N} \) as \( N \) random paths of the moving object over \( m \) frames. As shown in the weight updating equation (13), given the proposal density \( q(x_k^i|z_{k-m:k-1}, \mathbf{z}_k) \), a particle will be assigned higher weight if (i) the final destination is good, represented as having higher observation likelihood \( p(z_k^i|x_k^i) \) at time \( k \); (ii) the path (dynamics) is good, represented as with higher transition probability \( p(x_k^i|x_{k-m:k-1}^i) \).

### Table 1. Comparisons of first-order and mth-order particle filtering

<table>
<thead>
<tr>
<th>Tracking state</th>
<th>First-Order</th>
<th>Mth-Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>A description of tracking state</td>
<td>an object at one frame</td>
<td>an object’s moving path over ( m ) frames</td>
</tr>
<tr>
<td>Dynamics</td>
<td>( x_k = Ax_{k-1} + Bu_k )</td>
<td>( x_k = Ax_{k-1} + Bu_{k-1} + \cdots + Gu_{k-m} + Fv_k )</td>
</tr>
<tr>
<td>Weight updating</td>
<td>( w_k^i \propto \frac{p(x_k^i,z_k^i)q(x_k^i</td>
<td>z_{k-m:k-1}, \mathbf{z}_k)}{q(x_k^i</td>
</tr>
<tr>
<td>Estimation</td>
<td>( \hat{x}<em>k = \sum</em>{i=1}^{N} w_k^i \mathbf{x}_{k-m+1:k}^i )</td>
<td>( \hat{x}<em>{k-m+1:k} = \sum</em>{i=1}^{N} w_k^i \mathbf{x}_{k-m+1:k}^i )</td>
</tr>
<tr>
<td>Particles storage</td>
<td>( {\mathbf{x}<em>{k-m+1:k}^i}</em>{i=1}^{N} )</td>
<td>( {\mathbf{x}<em>{k-m+1:k}^i}</em>{i=1}^{N} )</td>
</tr>
</tbody>
</table>

### Table 2. Proposed high-order particle filtering tracking algorithm

- Initialize \( \{\mathbf{x}_0^i, w_0^i\}_{i=1}^{N} \) according to a prior distribution \( p(\mathbf{x}_0) \).
- For \( k = 1, 2, 3, \ldots \)
  - For \( i = 1, 2, \ldots, N \)
    - Keep particles \( \mathbf{x}_k^i \).
    - Draw particles \( \mathbf{x}_k^i \sim q(\mathbf{x}_k^i|\mathbf{x}_{k-m:k-1}^i, \mathbf{z}_k) \).
    - Calculate weight \( w_k^i \) as in (13).
  - End for \( i \)
  - Normalize weight \( w_k^i = w_k^i / \sum_{i=1}^{N} w_k^i \).
  - Estimate \( \hat{x}_k = \sum_{i=1}^{N} w_k^i \mathbf{x}_k^i \).
  - Resample \( \{\mathbf{x}_k^i\}_{i=1}^{N} \).
- End for \( k \)

3.3. High-Order Particle Filtering Tracking Algorithm

According to (14), ideally we can do particle filtering at step \( m \). For example, if \( m = 2 \), we can do estimate for frames \( k-1, k \) together and then comes frames \( k+1, k+2 \). However, in this case, the error will be accumulated while propagating for \( m \) frames and the resampling cannot be carried out timely to reduce the degeneracy problem [4]. Therefore, in applications we still consider one filtered estimate at each time step. Although we could get the estimated state matrix \( \mathbf{x}_{k-m+1:k} \), we only update \( \hat{x}_k \). From (14), we can obtain \( \hat{x}_k = \sum_{i=1}^{N} w_k^i \mathbf{x}_k^i \). In the above example, we do estimation for frames \( k-1, k \) together, but only update the object’s state for time \( k \) instead of \( k-1 \).

We then give the high-order particle filter tracking algorithm presented in Table 2. When \( m = 1 \), the tracking algorithm reduces to the generic particle filter in [4].

4. EXPERIMENTAL RESULTS

To demonstrate the improved performance of the proposed high-order particle filtering tracking algorithm, we perform experiments on synthetic sequences as well as real videos compared with traditional first-order particle filtering, i.e. particle filtering algorithm derived based on first-order Markov chain. We choose the proposal density to be the tran-
Fig. 3. Tracking results of the synthetic video Football: (a) 1st-order particle filtering tracking algorithm, (b) 2nd-order particle filtering tracking algorithm. A portion of the image is displayed for better presentation.

Fig. 4. The absolute error of the object’s center of the synthetic video Football.

The synthetic sequence Football has a football moves with a second-order dynamics in a challenging clutter environment with a resolution of 320 × 240. The dynamics of the object is predefined. Both of the methods use 20 particles per frame. Tracking results of the different algorithms are shown in Fig. 3. Figure 4 illustrates the absolute tracking error of the horizontal and vertical coordinates of the object’s center. Our proposed second-order algorithm outperforms the first-order algorithm while requiring about the same computation time (see Table 3).

The Boy video clip contains a boy running on campus. The resolution is 320 × 240 and the frame rate is 10 frames per second. The dynamics is learned from the training data and the number of particles used is 30 per frame. Fig. 5 illustrates the tracking results of the different algorithms. Our algorithm produces better tracking results compared to the first-order tracking algorithm.

We have implemented all of the comparative experiments independently in VC++ without code optimization on a 2.8 GHz Pentium IV PC. Compared with the first-order particle filtering, the mth-order tracking algorithm spends more time for keeping the particles and doing more additions while propagating samples. However, the extra time is negligible compared with other operations, e.g. calculating the local likelihood. We can check from Table 3 that the computation time per frame of the two algorithms is the same. The data has been averaged over 5 iterations on the Football sequence. However, we should notice that the high-order particle filtering tracking algorithm requires more storage memory for keeping the samples for past m frames.

5. CONCLUSION

In this paper, we present a novel approach for visual tracking using high-order Monte Carlo Markov chain. We use graphic models to obtain the conditional independence properties and derive the posterior density function for the mth-order hidden Markov model. Then we utilize Sequential Importance Sampling to estimate the posterior density and obtain the high-order particle filtering tracking algorithm. Simulation results demonstrate the better tracking performance of proposed high-order tracking algorithm compared with the traditional first-order particle filtering tracking algorithm. The proposed algorithm requires more memory for storage particles but improves tracking performance using the same computation time.

6. REFERENCES