Graph-Based Sequential Particle Filtering Framework for Articulated Motion Analysis

Jing Huang and Dan Schonfeld
Department of Electrical and Computer Engineering
University of Illinois at Chicago
Chicago, United States
Email: {jhuang39, dans}@uic.edu

Abstract—A general framework for sequential particle filtering on graphs is presented in this paper. We present two new articulated motion analysis and object tracking approaches: the graph-based sequential particle filtering framework for articulated object tracking and its hierarchical counterpart. Specifically, we estimate the interaction density by an efficient decomposed inter-part interaction model. To handle severe self-occlusion, we further formulate high-level inter-unit interaction and develop a hierarchical graph-based sequential particle filtering framework for articulated motion analysis. We rely on the proposed general framework of graph-based particle filtering for articulated motion analysis applications. The resulting experiments further demonstrate the superiority of our approach to tracking compared with existing methods.

Keywords—graphical models; particle filtering; occlusions; articulated motion analysis;

I. INTRODUCTION

Recent technological trends have required the deployment of articulated motion analysis applications driven by its wide applications such as human-computer interaction, patient rehabilitation, human activity analysis, computer animation, etc. Articulated motion analysis is a challenging task because of the exponentially increased computational complexity in terms of the degrees of freedom of the object and the severe image ambiguities incurred by frequent self-occlusions.

Many approaches have been studied to avoid the problems inherent in articulated motion tracking. Recently, Bayesian filtering framework has become very popular for tracking. It provides a recursive formulation of the posterior probability density function in dynamical systems. Analytical solutions for the optimal Bayesian filtering problem are known only for special cases including the linear and gaussian case (Kalman filter [1]). Particle filters [2] provide a general framework for estimating the probability density function of general non-linear and non-Gaussian systems. Most early efforts of articulated motion analysis took advantages of 2D and 3D object models [2]. A unified spatio-temporal articulated model was proposed by Lan and Huttenlocher [3]. Kalman filters have been employed by many researchers to combat occlusions in articulated object tracking [4]. Sequential Monte Carlo method or particle filter was demonstrated to be efficient for object tracking in clutter [2] and has also been introduced for articulated motion analysis. Deutsher et al. [5] modified the Condensation algorithm [2] by an annealed particle filter. Choo et al. [6] described a filter that used hybrid Monte Carlo to obtain efficient samples in high-dimensional spaces. Chang et al. [7] proposed an appearance-based particle filter for articulated hand tracking. The successful application of particle filtering was limited to situations where the dimension of the joint state is relatively small. For high-dimensional state spaces, many algorithms become computational inefficient.

Recently, probabilistic graphical models have been used to facilitate the analysis of high dimensionality signal processing problems. It provides a more simple and distinct way to visualize the structure of the probability model. For example, variational analysis methods [8] [9] are generally used to obtain approximate inference for loopy Markov networks. They provide lower bounds of the approximation as a theoretical benefit. In contrast to variational analysis methods, loopy belief propagation [10] often converges and when it do, it gives a better approximation. Particle filtering on general graphs is first proposed by [11]. They split the general graph into multiple cycle-free subgraphs and apply the filtering algorithm on cycle-free graphs in a distributed way.

In this paper, we present a general framework for sequential particle filtering on graphs and use the proposed general framework for articulated motion analysis applications. Our main contributions are as follows: (i) We first introduce an efficient decomposed inter-part interaction model and apply graph-based sequential particle filtering framework for articulated object analysis; (ii) To handle severe self-occlusion, we further formulate high-level inter-unit interaction; (iii) A novel hierarchical graph-based sequential particle filtering framework is proposed for articulated motion analysis.

II. GRAPH-BASED SEQUENTIAL PARTICLE FILTERING FRAMEWORK

In the area of articulated object tracking, the motion of neighboring part has some inherent constraints. A good example of such constraints appears in the human body. It can be observed that there is a common relationship
among arms and legs. Left arm moves forward while left leg moves backward. This information can be used to solve self-occlusion problem. We first introduce a decomposed inter-part interaction model for articulated motion analysis. Then a graph-based particle filtering framework is formulated in this section.

A. Graphical Models for Articulated Object Representation

An articulated object can be represented by a graphical model in Fig. 1. It has two layers: the hidden state layer (circle nodes) and the observation layer (square nodes). Each circle node corresponds to a part of the articulated object. For instance, considering a human body, a part can be a torso, or a thigh, etc. The undirected links represent physical constraints among different articulated parts. Each individual part is associated with its observation. The directed link from a part’s state to its associated observation represents the local observation likelihood. Instead of using the joint state representation for the whole articulated object, we denote the state of each part at time $t$ by $x_{i,t}$, where $i = 1, 2, ..., M$ is the index of parts. In our implementation, the state $x_{i,t}$ is chosen as $x = (cx, cy, b, \Theta)$, where $(cx, cy)$ is the center point; $b$ is half of the length of the rectangle; $\Theta$ is the rotation angle of state around the center point with respect to the Y axis. The ratio of the length and width of the rectangle is held constant equal to its value obtained in the first frame. We denote the observation of $x_{i,t}$ by $z_{i,t}$.

The inter-part interaction density $p(x_{j,t} | x_{i,t})$ models the constraints between analyzed part $i$ and its neighboring part $j$. Estimation of this density should adapt to different applications and is usually critical in practical implementation. To avoid high computational requirements for a joint state representation model, we develop an efficient decomposed inter-part interaction model based on [12]. It can be observed that the relative locations and poses of two adjacent parts are independent. Therefore, by temporarily discarding the time index, we have

$$p(x_{j,t} | x_{i,t}) = p(cx_j \mid cy_i, b_i, \Theta_j | cx_i, cy_i, b_i, \Theta_i) = p(cx_j, cy_j \mid cx_i, cy_i)p(\Theta_j \mid \Theta_i)p(b_j \mid b_i)$$ (1)

Where we assume that the size of an object part is not influenced by its neighboring parts. Without considering the size relation between two parts, $p(b_j \mid b_i)$ becomes uniformly distributed. Thus, we can further simply (1) to be

$$p(x_{j,t} | x_{i,t}) \propto p(c^j \mid c^i)p(\Theta^j \mid \Theta^i)$$ (2)

Where $c^i = (cx^i, cy^i), c^j = (cx^j, cy^j)$ are the coordinates of the center points, $p(c^j \mid c^i)$ models the location interaction of two adjacent parts. We adopt a "spring-joint" model similar to [12] for $p(c^j \mid c^i)$.

$$p(c^j \mid c^i) = \frac{1}{2\pi|\Sigma_e|^{1/2}} \exp\{-1/2(e^j - e^i)^T\Sigma_e^{-1}(e^j - e^i)\}$$ (3)

Where $\Sigma_e$ is the covariance matrix of this bivariate normal distribution. In (1), $p(\Theta^j \mid \Theta^i)$ models the pose relation of two adjacent parts. It can be estimated either by some prior knowledge in particular applications, or by learning from training data.

B. Posterior Density Propagation And Sequential Importance Sampling

In our graph representation, each node represents a hidden state $x_i$ and is linked to an observation $z_i$. We use $V$ to denote the set of hidden states in graph $G$, i.e. $V = \{x_0, x_1, x_2, \ldots\}$. Set $V$ could be partitioned into disjoint sets, called layers, $\{V_l\}_{l=0}^L$. The number of elements in $V_l$ is $K(l)$. Let us use set $V_0$ to denote the set of nodes that have no parents. Furthermore, we use the notation $v_{m,l}$ to denote the $m$th node in $V_l$, $m = 1, 2, ..., K(l)$. The parents of $v_{m,l}$ can be denoted as $Pa(v_{m,l})$. It could be easily seen that $Pa(v_{m,l}) \subseteq V_{0,l-1}$. The observation associated with $v_{m,l}$ is denoted as $o_{m,l}$, and $O_l = \{o_{m,l}, m = 1, 2, ..., K(l)\}$. We denote the order of node $v_{m,l}$ as $S(v_{m,l})$. Although therefore the order of the nodes in a cycle-free graph will have many combinations, we only use one predetermined order.

Take Fig. 2 as an example, $V = \{x_0, x_1, x_2, x_3, x_4, x_5\}$, $V_0 = \{x_0\}$, $V_1 = \{x_1, x_2\}$, $V_2 = \{x_3\}$, $V_3 = \{x_4\}$ and $V_4 = \{x_5\}; v_{1,0} = x_0, v_{1,1} = x_1, v_{2,1} = x_2, v_{1,2} = x_3, v_{1,3} = x_4, v_{1,4} = x_5$. $Pa(v_{1,2}) = \{x_1, x_2\}$, one possible order of the cycle-free graph in Fig. 2 is $S(x_0) = 1, S(x_3) = 4, S(x_5) = 6$.

The nodes in set $V_0$ do not have any parents, and therefore the conditional probability is given as prior $p(v_{m,0})$. For the node $v_{m,l}$ in $V_l$, we formulate the conditional density $p(v_{m,l} | V_{0,l-1} o_{m,l}, O_{1:l-1})$ as
Figure 2. Example of a directed cycle-free graph (the observation layer is omitted).

\[
p(v_{m,t}, V_{0:t-1} | o_{m,t}, O_{1:t-1}) \\ \propto p(o_{m,t} | v_{m,t}) p(v_{m,t} | P_a(v_{m,t})) p(V_{0:t-1} | O_{1:t-1})(4)
\]

We further use sequential importance sampling technique [2] again as the paradigm. We denote \( \{v_{m,t}, V_{0:t-1}\}^N_{i=1} \) as a random measurement that characterizes the posterior density in (4), where \( \{v_{m,t}, V_{0:t-1}\}^i \) is a set of N particles with associated weights \( w^i_{S(v_{m,t})} \). The weights are normalized such that \( \sum_{i=1}^{N} w^i_{S(v_{m,t})} = 1 \).

Therefore,

\[
p(v_{m,t}, V_{0:t-1} | o_{m,t}, O_{1:t-1}) \approx \sum_{i=1}^{N} w^i_{S(v_{m,t})} \delta (v_{m,t} - v^i_{m,t}) (5)
\]

where the weights are given by

\[
w^i_{S(v_{m,t})} \propto \frac{p(v_{m,t} | v^i_{m,t}, V_{0:t-1}) q(v^i_{m,t} | o_{m,t}, O_{1:t-1})}{q(v^i_{m,t} | o_{m,t}, V_{0:t-1})} (6)
\]

By using the conditional independence properties [13], we obtain \( q(v_{m,t} | V_{0:t-1}, o_{m,t}, O_{1:t-1}) = q(v_{m,t} | o_{m,t}, P_a(v_{m,t})q(V_{0:t-1}) | O_{1:t-1}) \). The weight update equation can be further obtained as follows:

\[
w^i_{S(v_{m,t})} \propto w^i_{S(v_{m,t})} \frac{p(o_{m,t} | v^i_{m,t}) p(v^i_{m,t} | P_a(v_{m,t}))}{q(v^i_{m,t} | P_a(v_{m,t}))} (7)
\]

In (7), \( p(o_{m,t} | v^i_{m,t}) \) is the likelihood, and \( p(v^i_{m,t} | P_a(v_{m,t})) \) captures the relationship between the particles of the current node and its parents, and \( q(v^i_{m,t} | P_a(v_{m,t})) \) is the proposal density.

In applications, we still consider one filtered estimate at each time step. Therefore, we have

\[
p(v_{m,t} | o_{m,t}, O_{1:t-1}) \approx \sum_{i=1}^{N} w^i_{S(v_{m,t})} \delta (v_{m,t} - v^i_{m,t}) (8)
\]

The graphical model of articulated object tracking is given in Fig. 3. The notation \( x_{i,t} \) represents the hidden state of part \( i \) at time \( t \), and \( z_{i,t} \) is the observation associated to it. The undirected link between circle nodes represents the interaction among parts, e.g., self-occlusion. When a graph has cycles, e.g., parts 1, 2 and 4, we could split the graph into multiple cycle-free subgraphs by following the approach presented in [11] and apply the particle filtering algorithm on cycle-free graphs in a distributed way.

From (7), (8) and the graphical model of articulated object tracking, we get

\[
p(x_{i,t} | z_{i,t}, z_{1:{S(x_{i,t})-1}}) \approx \sum_{n=1}^{N} w^n_{i,t} \delta (x_{i,t} - x^n_{i,t}) (9)
\]

The weights are normalized such that \( \sum_{n=1}^{N} w^n_{i,t} = 1 \). The weight update equation can be further obtained as follows:

\[
w^n_{i,t} \propto w^n_{S(x_{i,t})} \frac{p(z_{i,t} | x^n_{i,t}, P_a(x_{i,t}))}{q(x^n_{i,t} | P_a(x_{i,t}))} (10)
\]

In (10), \( p(x^n_{i,t} | P_a(x_{i,t})) \) captures the relationship between the particles of the current node and its parent. Depending on whether there are interactions among different parts of articulated objects at the same time, \( p(x^n_{i,t} | P_a(x_{i,t})) \) could be divided into two categories as follows:

1. If there are no interactions among different parts at the same time (e.g., part 3), \( p(x^n_{i,t}, P_a(x_{i,t})) = p(x^n_{i,t} | x^n_{i,t-1}) \) is determined by the dynamics of the isolated part, which is usually considered as a random walk or learned from training data. Therefore, (10) becomes

\[
w^n_{i,t} \propto w^n_{S(x_{i,t})} \frac{p(z_{i,t} | x^n_{i,t}, P_a(x_{i,t}))}{q(x^n_{i,t} | P_a(x_{i,t}))} (11)
\]

2. If different parts of articulated object interact at time \( t \) (e.g., parts 1, 2, 4 and 5), we have \( p(x^n_{i,t} | P_a(x_{i,t})) = p(x^n_{i,t} | x^n_{i,t-1}, \{ln(x_{i,t})\}^n) \), where \( ln(x_{i,t}) \) represents the interacting parts of \( x_i \) at time \( t \); e.g., \( p(x^n_{i,t} | x^n_{i,t-1}, x^n_{i,t-1}, \{ln(x_{i,t})\}^n) \) described in Fig. 3. Because interacting parts must compete for limited observations, we therefore model
the probability as $p(x^n_{i,t-1}|\{x^n_{i,t}\}) \propto p(x^n_{i,t}|\{x^n_{i,t-1}\}) \phi(x^n_{i,t}, \{ln(x_{i,t})\})$. We define 
\[
\phi(x^n_{i,t}, \{ln(x_{i,t})\}) = \prod_{x_{j,t} \in \ln(x_{i,t})} \{ \sum_{w_{j,t} \in S_{x_{j,t}}} w(x^n_{j,t})p(x^n_{j,t}|x^n_{i,t}) \}
\]
where $N_j$ is the total number of samples of part $j$. The density $p(x^n_{j,t}|x^n_{i,t})$ models the interaction between two neighboring parts’ samples $x^n_{j,t}$ and $x^n_{i,t}$. $w(x^n_{j,t})$ acts as a weight to the associated interaction. They work together to constrain the neighboring parts and prevent different parts of the articulated object from separating over time. Therefore, (10) becomes

\[
w^n_{i,t} \propto w^n_{S(x_{i,t})}^{-1} p(z^n_{i,t}|x^n_{i,t}) p(x^n_{i,t}|x^n_{i,t-1}) \times \prod_{x_{j,t} \in \ln(x_{i,t})} \{ \sum_{w_{j,t} \in S_{x_{j,t}}} w(x^n_{j,t})p(x^n_{j,t}|x^n_{i,t}) \}
\]

III. HIERARCHICAL GRAPH-BASED PARTICLE FILTERING FRAMEWORK

The interaction inside an articulated object lies not only in the adjacent parts but also some "high-level" nonadjacent "part groups". For clarity, we define a group of parts as a unit, which is denoted by $X_{I,t}$, where $I = 1, \ldots, M’$; $M’$ is the total number of units. For instance, each limb of a human body contains two parts and can, thus, be regarded as a unit. Similar to the model in Fig. 1, but considering the "high-level" unit interaction as well, we represent the same articulated object in Fig. 1 by a hierarchical graphical model as illustrated in Fig. 4. Compared with the model in Fig. 1, the difference of this hierarchical model is that it introduces a high-level layer containing big blue ellipse nodes and red curve links. Each big ellipse node corresponds to a unit of the articulated object. The undirected curve links between units represent "high-level" interaction. We denote the related neighboring units of $X_{I,t}$ by $\ln(I)$, the joint state of all these related neighboring units by $X_{\ln(I),t} = \{X_{K,t}, K \in \ln(I)\}$, and the corresponding observations by $Z_{\ln(I),t} = \{Z_{K,t}, K \in \ln(I)\}$. We assume that $p(X_{\ln(I),t}|x_{i,t}) = p(X_{\ln(I),t}|X_{I,t})$. This assumption assumes all the parts $x_{i,t} \in X_{I,t}$ share the same "relation" with the neighboring units $X_{\ln(I),t}$, which is the interaction between high-level units $X_{\ln(I),t}$ and $X_{I,t}$.

Compared with the graph-based particle filtering framework in the previous section, hierarchical framework in this section introduces an additional high-level inter-unit weighting factor. We can also use the sequential Monte Carlo method [2] to approximate the conditional density propagation rule derived in the previous section. Therefore, (10) becomes
Table 1

<table>
<thead>
<tr>
<th>Method</th>
<th>GBPF</th>
<th>MIPF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average MSE</td>
<td>2.08</td>
<td>9.67</td>
</tr>
<tr>
<td>CPU time</td>
<td>42.6</td>
<td>19.7</td>
</tr>
</tbody>
</table>

IV. EXPERIMENTAL RESULTS

In this section, we report some of the experimental results. The tracking performance of the proposed two articulated motion analysis methods are compared both qualitatively and quantitatively with the multiple independent particle filtering (MIPF) [2], mean field Monte Carlo (MFMC) [14], respectively. For all methods, we use 50 particles for each part. In our simulations, the tracked parts are identified by the user. The number of tracked parts in articulated object is therefore pre-determined, and the parts are assumed to have a uniform prior distribution. Different colors of the rectangular are used to label the parts. The dynamics of the object is considered as a random walk, with the noise variance the same for all comparative methods.

The Boy sequence is from a video taken by a hand-held camcorder, which is common for a lot of activities nowadays. The video Boy contains a boy moving his arms. We apply graph-based particle filtering (GBPF) framework to the boy in the video using 50 particles per part. For comparison, we implemented MIPF [2] as baseline method. As shown in Fig. 5, the figures in the first row report the tracking results of GBPF framework. By exploiting the physical adjacent constraints of the human body, the GBPF improves the tracking performance in that the connections among parts are preserved well. The figures in the second row report the tracking performance for MIPF [2]. Note that our proposed GBPF can track each part of the object in most frames for irregular motions, while MIPF [2] loses track after self-occlusion. Table I compares the average MSE and computational time of the two methods in Fig. 5. From Table I, we indeed observe that the computation time required for the proposed GBPF is higher than the MIPF [2], simply because we introduce interaction weighting factor among different parts of the object. Furthermore, we compare the horizontal and vertical coordinates of two selected parts’ trajectories (Yellow and Purple rectangles) from the Boy sequence for frame 50 to frame 150. Fig. 6 illustrates that the selected part loses track after fast motion and self-occlusions in MIPF [2]. Note that the proposed GBPF is more accurate than MIPF [2].

The Walking sequence contains a person walking forward inside a lab. The similar color of the torso and arms, and the frequent severe self-occlusions among limbs make it difficult for articulated motion analysis. We illustrate the sample result frames of our proposed two graph-based framework, MIPF [2] and MFMC [14] in Fig. 7. MFMC [14] outperforms MIPF [2] since it keeps the connection among the parts. However, it could not produce satisfactory results when self-occlusions present. Compared with MFMC [14], our proposed graph-based particle filtering (GBPF) framework improves the performance in that the connections among parts are preserved well. This is because our proposed framework uses separated interaction models for the location and rotation. By handling the high-level interaction among arms and legs and using a learned model of limb poses, hierarchical graph-based particle filtering (HGBPF) framework gives the best results. Table II compares the average MSE and computational time of the four methods in Fig. 7. From Table II, we indeed observe that MFMC [14] improves the tracking performance compared with MIPF [2]. Moreover, the proposed two methods model the interaction constraints of an articulated object more efficiently compared with MFMC [14] and therefore achieve more accurate tracking results. (a), (b), (c) and (d) in Table II represent MIPF [2], MFMC [14], proposed graph-based framework.
Figure 7. Tracking results of the Walking sequence for frames 16, 32, 53 and 65: (a) multiple independent particle filters (MIPF) [2]; (b) mean-field Monte Carlo (MFMC) [15]; the proposed graph-based particle filtering (GBPF) method; and (d) proposed hierarchical graph-based particle filtering (HGBPF) method.

Table II

<table>
<thead>
<tr>
<th></th>
<th>(a)</th>
<th>(b)</th>
<th>(c)</th>
<th>(d)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average MSE</td>
<td>6.08</td>
<td>3.67</td>
<td>2.12</td>
<td>1.55</td>
</tr>
<tr>
<td>CPU time</td>
<td>28.6</td>
<td>36.1</td>
<td>45.7</td>
<td>49.6</td>
</tr>
</tbody>
</table>

and its hierarchical counterpart, respectively.

V. CONCLUSION

In this paper, we presented a general framework for sequential particle filtering on graphs. We presented two new articulated motion analysis and object tracking approaches: the graph-based sequential particle filtering framework for articulated object tracking and its hierarchical counterpart. Specifically, we estimated the interaction density by an efficient decomposed inter-part interaction model. To handle severe self-occlusion, we further formulated high-level inter-unit interaction and developed a hierarchical graph-based sequential particle filtering framework for articulated motion analysis. We utilized the proposed general framework of graph-based particle filtering for articulated motion analysis. The resulting experiments further demonstrated the superiority of our approach to tracking compared with existing methods.

REFERENCES


