Unsupervised Learning of Human Motion

Yang Song, *Member*, *IEEE*, Luis Goncalves, *Member*, *IEEE*, and Pietro Perona, *Member*, *IEEE*

Abstract—An unsupervised learning algorithm that can obtain a probabilistic model of an object composed of a collection of parts (a moving human body in our examples) automatically from unlabeled training data is presented. The training data include both useful "foreground" features as well as features that arise from irrelevant background clutter—the correspondence between parts and detected features is unknown. The joint probability density function of the parts is represented by a mixture of decomposable triangulated graphs which allow for fast detection. To learn the model structure as well as model parameters, an EM-like algorithm is developed where the labeling of the data (part assignments) is treated as hidden variables. The unsupervised learning technique is not limited to decomposable triangulated graphs. The efficiency and effectiveness of our algorithm is demonstrated by applying it to generate models of human motion automatically from unlabeled image sequences, and testing the learned models on a variety of sequences.

Index Terms—Unsupervised learning, human motion, decomposable triangulated graph, probabilistic models, greedy search, EM algorithm, mixture models.

1 Introduction

Automatic detection and tracking of people, and analysis of their motion, actions, and activities, are important areas in computer vision with potential applications to medicine, entertainment, and security. To this end, a number of models of human motion have been proposed in the literature [14]. "Strong" models represent explicitly the kinematics and dynamics of the human body [25], [24], [4], [15], [16], [5], [31], while "weak" models represent its phenomenological spatio-temporal appearance [23], [35], [27], [26]. Strong models have the advantage of incorporating more information and, in principle, tolerate lower signal-to-noise ratios and be allowed to reconstruct 3D body pose and motion from 2D images. Weak models allow the representation of motion patterns, where physical and geometrical models are not easy to obtain (e.g., loose clothing, bodies of unknown dimensions) and may therefore be more practical for image-based tasks, such as detection and recognition. Another potential advantage of weak models is that they are, in principle, cheaper to reprogram to represent different complex motions (whether human or not) since a detailed analysis of the geometry and physics of the moving object is not needed. It is therefore useful to develop methods to train weak models from image sequences with minimal user assistance.

We propose a method for learning weak models automatically from image sequences; more specifically, we focus here on probabilistic models proposed by Song et al. [27], [26]. The human motion is modeled by the joint probability density function of the position and velocity of a collection of body parts. The probabilistic conditional independence structure of body parts, encoded by a decomposable triangulated graph, is such that it allows efficient detection and labeling of the body. Structure learning of graphical models has been previously studied by a number of authors [7], [12], [20], [22], [13]. The main contribution of this paper, apart from the specifics of the application, is that our method is unsupervised: it is based on *unlabeled* training data. The training sequence contains a number of bottom-up features (Tomasi and Kanade points [30] in our implementation) which are *unlabeled*, i.e., we do not know which features are associated to the body, which to background clutter, which features correspond to which features across image frames, and which features are the most informative for a given task. The learning algorithm must therefore choose a number of useful features as body parts, establish their correspondence across frames, determine the underlying probabilistic independence structure, and estimate the parameters of the probability density function. One added generalization of our setting is that the features corresponding to body parts are not required to be present in all frames (neither during learning nor during detection and labeling).

In Section 2, we summarize the main facts about decomposable triangulated probabilistic models and how to use them to perform human motion detection and labeling. In Section 3, we address the learning problem when the training features are *labeled*, i.e., the parts of the model and the correspondence between the parts and observed features are known. In Section 4, we address the learning problem when the training features are *unlabeled*. In Section 5, we introduce the concept of mixtures of decomposable triangulated models and extend the unsupervised learning algorithm to the mixture model. In Section 6, we present some experimental results.

- *Y. Song is with Fujifilm Software, Inc., 1740 Technology Dr., Suite 490, San Jose, CA 95110. E-mail: ysong@fujifilmsoft.com.*
- *L. Goncalves is with Idealab, 130 W. Union Street, Pasadena, CA 91103. E-mail: luis.goncalves@idealab.com.*
- *P. Perona is with the Department of Electrical Engineering, 136-93, California Institute of Technology, Pasadena, CA 91125. E-mail: perona@.caltech.edu.*

Manuscript received 27 Sept. 2002; revised 2 Mar. 2003; accepted 17 Mar. 2003.
Recommended for acceptance by V. Pavlovic.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 118243.

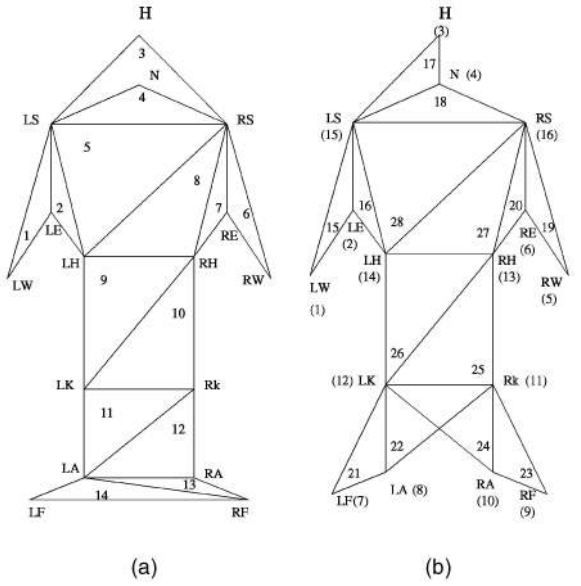


Fig. 1. Two decompositions of the human body into triangles [27]. "L" and "R" in label names indicate left and right. H: head, N: neck, S: shoulder, E: elbow, W: wrist, H: hip, K: knee, A: ankle, and F: foot. In (a), the numbers inside triangles give one order in which the vertices can be deleted. In (b), the numbers in brackets show one elimination order.

2 Decomposable Triangulated Graphs

We use *decomposable triangulated* graphs[1] to model the probabilistic conditional independence structure of body parts. A decomposable triangulated graph [1] is a collection of cliques[2] of size three, where there is an elimination order of vertices such that 1) when a vertex is deleted, it is only contained in one triangle (we call it a free vertex) and 2) after eliminating one free vertex and the two edges associated with it, the remaining subgraph is again a collection of cliques of size three until only one triangle is left.

2.1 Detection and Labeling

In [27], [26], decomposable triangulated graphs are used to model the conditional independence of body parts, and dynamic programming is used for efficient detection and labeling. Fig. 1 shows two decomposable graphs of the whole body used in [27], along with one order of successive elimination of the cliques. For the sake of making this paper self-contained, we summarize the main results in this section.

Let $\mathcal{S}_{body} = \{S_1, S_2, \ldots, S_M\}$ be the set of M body parts. For example, S_1 denotes the left wrist, S_M is the right foot, etc., X_{S_i}, $1 \le i \le M$, is the measurement for S_i, the position and velocity of body part S_i in our application. We model the pose and motion of the body by means of a probability density function $P_{\mathcal{S}_{body}}$.

Let $\overline{X} = [X_1, \ldots, X_N]$ be a vector of measurements (each $X_i, i = 1, \ldots, N$ is a vector describing position and velocity of point i). For clarity of description, we first assume that there are no missing body parts and no clutter. In this case, $N = M$. Let $\overline{L} = [L_1, \ldots, L_N]$ be a vector of labels, where $L_i \in \mathcal{S}_{body}$ is the label of X_i. The best labeling of the scene is a vector $\overline{L}^*$, such that the posterior probability of the labeling given the observed data, $P(\overline{L}^*|\overline{X})$, is maximized over all possible label vectors $\overline{L}$. By Bayes' rule and equal priors assumption,[3] we have

$$\overline{L}^* = \arg\max_{\overline{L} \in \mathcal{L}} P(\overline{X}|\overline{L}), \tag{1}$$

where $\mathcal{L}$ is the set of all possible labelings.

If the conditional independence of body parts $\mathcal{S}_{body}$ can be represented as a decomposable triangulated graph, the joint probability density function $P_{\mathcal{S}_{body}}$ can be decomposed into,

$$\begin{aligned} &P_{\mathcal{S}_{body}}(X_{S_1}, X_{S_2}, \ldots X_{S_M}) \\ &= \textstyle\prod_{t=1}^{T-1} P_{A_t|B_tC_t}(X_{A_t}|X_{B_t}, X_{C_t}) \cdot P_{A_TB_TC_T}(X_{A_T}, X_{B_T}, X_{C_T}), \end{aligned} \tag{2}$$

where $A_i, B_i, C_i \in \mathcal{S}_{body}, 1 \le i \le T = M - 2$, and (A_1, B_1, C_1), $(A_2, B_2, C_2), \ldots, (A_T, B_T, C_T)$ are the cliques. $(A_1, A_2, \ldots, A_T)$ gives one elimination order, and B_i and C_i, $1 \le i \le T$ are the two vertices connected to A_i when it is deleted. $\{A_1, A_2, \ldots, A_T\} \cup \{B_T, C_T\}$ is the set of body parts, i.e., $\{A_1, A_2, \ldots, A_T, B_T, C_T\} = \mathcal{S}_{body}$. A dynamic programming algorithm [1], [27] can be used to compute the maximum likelihood $P(\overline{X}|\overline{L})$,

$$\begin{aligned} &\max_{\overline{L} \in \mathcal{L}} P(\overline{X}|\overline{L}) \\ &= \max_{X_{S_1}, X_{S_2}, \ldots, X_{S_M}} P_{\mathcal{S}_{body}}(X_{S_1}, X_{S_2}, \ldots X_{S_M}) \\ &= \max_{X_{A_1}, \ldots, X_{A_T}, X_{B_T}, X_{C_T}} \textstyle\prod_{t=1}^{T-1} P_{A_t|B_tC_t}(X_{A_t}|X_{B_t}, X_{C_t}) \\ &\quad \cdot P_{A_TB_TC_T}(X_{A_T}, X_{B_T}, X_{C_T}) \qquad (3) \\ &= \max_{X_{A_T}, X_{B_T}, X_{C_T}} (P_T(X_{A_T}, X_{B_T}, X_{C_T}) \\ &\quad \cdot \max_{X_{A_{T-1}}} (P_{T-1}(X_{A_{T-1}}|X_{B_{T-1}}, X_{C_{T-1}}) \cdots \\ &\quad \cdot \max_{X_{A_2}} (P_2(X_{A_2}|X_{B_2}, X_{C_2}) \cdot \max_{X_{A_1}} P_1(X_{A_1}|X_{B_1}, X_{C_1})))). \qquad (4) \end{aligned}$$

The equal sign from (3) to (4) is a key step in achieving computational efficiency: dynamic programming, which is from the decomposable property of the graph [1], [27]. The complexity of the dynamic programming algorithm is on the order of $M * N^3$. In [27], [26], the algorithms are extended to handle occlusion (some body parts missing) and clutter (some points not belonging to the body).

Detection consists of deciding whether a human body is present. We propose two strategies [28], [27], [26]: one is to threshold the best labeling found as above, the so-called winner-take-all strategy, and the other is to sum over all the hypothesis labelings, which can be computed efficiently using another dynamic programming procedure with the same computational complexities (using the "sum" operator instead of the "max" operator). For simplicity, we use the first strategy in this paper.

1. For general graphical models, the term *decomposable* and the term *triangulated* have their own meanings (they are actually equivalent properties [21]). Here, we use the term *decomposable triangulated* specifically for the graph type defined in this paragraph.

2. A clique is a maximal subset of vertices, any two of which are adjacent.

3. The equal priors assumption is reasonable when we have little knowledge about the labelings. We use it mainly due to its computational simplicity. There are also other ways to model the prior $P(\overline{L})$. For instance, if we have some prior knowledge on the number of background (clutter) points, $P(\overline{L})$ can be more precisely estimated. In [32], the number of clutter points is modeled with a Poisson distribution. However, it is hard to include this kind of global term in the dynamic programming algorithm.

2.2 Decomposable Triangulated Graphs and General Graphical Models

For general graphical models, the labeling problem is the most-probable-configuration problem on the graph and can be solved through max-propagation on junction trees [18], [21], [28]. The dynamic programming algorithm [2] and the max-propagation algorithm essentially have the same order of complexity which is determined by the maximum clique size of the graph.

The maximum clique size for a decomposable triangulated graph is three. Since any graph with maximum clique size equal to or less than three can be transformed into a decomposable triangulated graph by adding edges, decomposable triangulated graphs are the most powerful, or for any probability distribution, can provide the most accurate approximation, among all the graphs with less or similar computational cost. Another type of widely used graphs in modeling conditional (in)dependence is trees [7], [22], [17], whose maximum clique size is two. There exist efficient algorithms [8] to obtain the maximum spanning tree. Therefore, trees have computational advantages over decomposable triangulated graphs. But, decomposable triangulated graphs are more suitable for our application because they have better graph connectivity in dealing with occlusion [28]. With a tree graph [11], if there is a single occlusion, the detection result may be split into two or more separate components, whereas with a triangulated graph, even if two adjacent parts (vertices) are occluded, the detection may still be connected.

3 Supervised Learning of the Graph Structure

In this section, we explore learning graph structure from labeled data, i.e., with known correspondence between the parts and the observed features (e.g., data from a motion capture system [27]). This will be used as foundations for dealing with unlabeled training data in Section 4. Unfortunately, the problem of finding the optimal decomposable triangulated graph is NP-hard.[4] However, we can hope to find efficiently approximate solutions that are close to the optimal. To this end, we study a greedy algorithm based on the optimization criterion presented in Section 3.1.

3.1 Optimization Criterion

Our goal is to find the decomposable triangulated graph that can best describe the data. The notation for the set of body parts and the decomposition of the joint probability density function into decomposable triangulated graphs are defined in Section 2.1 and (2).

Suppose $\mathcal{X} = \{\overline{X}^1, \overline{X}^2, \ldots, \overline{X}^N\}$ is a set of i.i.d samples from a probability density function of M body parts, where $\overline{X}^n = (X^n_{S_1}, \ldots, X^n_{S_M})$, $1 \le n \le N$, and $X^n_{S_i}$, $1 \le i \le M$ is the measurements of body part S_i. We call such $\overline{X}^n$ **labeled data**,[5] since the correspondence between the body parts and measurements is known. In a maximum-likelihood setting, we want to find the decomposable triangulated graph G, such that $P(G|\mathcal{X})$ is maximized over all possible such graphs. $P(G|\mathcal{X})$ is the probability of graph G being the "correct" one given the observed data $\mathcal{X}$. By Bayes' rule, $P(G|\mathcal{X}) = P(\mathcal{X}|G)P(G)/P(\mathcal{X})$. Therefore, if we make the simplifying assumption that the priors $P(G)$ are equal for different decomposable triangulated graphs, then our goal is to find the structure G which can maximize $P(\mathcal{X}|G)$. By (2), $P(\mathcal{X}|G)$ can be computed as follows [9], [7], [12], [20], [22]:

$$\begin{aligned}
\log P(\mathcal{X}|G) &= \sum_{n=1}^{N} \log P(\overline{X}^n|G) \\
&= \sum_{n=1}^{N} \Big(\sum_{t=1}^{T-1} \log P(X^n_{A_t}|X^n_{B_t}, X^n_{C_t}) \\
&\qquad + \log P(X^n_{A_T}, X^n_{B_T}, X^n_{C_T}) \Big) \qquad (5) \\
&\cong -N \cdot \sum_{t=1}^{T-1} h(X_{A_t}|X_{B_t}, X_{C_t}) - N \cdot h(X_{A_T}, X_{B_T}, X_{C_T}) \qquad (6) \\
&= -N \cdot \sum_{t=1}^{T} h(X_{A_t}|X_{B_t}, X_{C_t}) - N \cdot h(X_{B_T}, X_{C_T}), \qquad (7)
\end{aligned}$$

where $h(\cdot)$ is differential entropy or conditional differential entropy [9] (we consider continuous random variables here). Equation (6) is an approximation which converges to equality for $N \to \infty$ due to the weak Law of Large Numbers. We want to find the decomposition $(A_1, B_1, C_1), (A_2, B_2, C_2), \ldots, (A_T, B_T, C_T)$ such that $\log P(\mathcal{X}|G)$ can be maximized.

3.2 Greedy Search

The search for the optimal decomposable triangulated graph is an NP-hard problem. This section develops a greedy algorithm to grow the graph by the property of decomposable graphs. We start from a single vertex, and add vertices one by one each time maximizing (7). For each possible choice of C_T (the last vertex of the last triangle), find the B_T which maximizes $-h(X_{B_T}, X_{C_T})$, then get A_T, the vertex (part) that can maximize $-h(X_{A_T}|X_{B_T}, X_{C_T})$. Add edges (A_T, B_T) and (A_T, C_T) to the graph. The next vertex is added to the existing graph by choosing the best child of all the edges (legal parents) of the existing graph. This is repeated until all the vertices are added to the graph. For each choice of C_T, one such graph can be grown, so there are M candidate graphs. The final result is the graph with the highest $\log P(\mathcal{X}|G)$ among the M graphs.

The above algorithm is efficient. The number of possible choices for C_T is M, the number of choices for B_T is $M-1$; for stage t, $M-2 = T \ge t \ge 1$, the number of edges in the graph obtained so far (legal parents) is $2*(T-t)+1$ and the number of vertices to be added to the graph (legal children) is t. Therefore, the total search cost is $M*(M-1+\sum_t((2*(T-t)+1)*t))$, which is on the order of M^4. There is, of course, no guarantee that the global optimal solution will be found. The effectiveness of the algorithm will be explored through experiments.

There are also other approximate ways to build the model. For example, we can add edges to a maximum spanning tree

4. From Section 2, we know that the search for the optimal decomposable triangulated graph is equivalent to the search for the optimal graph with tree-width not greater than three. It is proven that the latter problem is NP-hard [6], [29]. Therefore, the search of the optimal decomposable triangulated graph is NP-hard.

5. Note that $\overline{X}^n$ in Section 3 is different from other sections. Here, $\overline{X}^n$ is a sample from a probability distribution of M body parts. It only includes measurements of body parts with known correspondence. In other sections, it denotes the observed measurements that include body parts and background clutter.

(MST), but it has been shown to be inferior to the greedy search for our application (see [28] and Fig. 3 for more details).

3.3 Computation of Differential Entropy-Translation Invariance

In the greedy search algorithm in Section 3.2, we need to compute $h(X_{A_t}|X_{B_t}, X_{C_t}) = h(X_{A_t}, X_{B_t}, X_{C_t}) - h(X_{B_t}, X_{C_t})$, $1 \le t \le T$. Although our algorithm could work with any choice of probability distribution, we chose to model the joint density of body parts with a Gaussian distribution since it is mathematically convenient and experiments indicate that it is a reasonable assumption. Thus, the differential entropy can be computed by $\frac{1}{2}\log(2\pi e)^n|\Sigma|$, where n is the dimension and Σ is the covariance matrix [9].

In our applications, position and velocity are used as measurements for each body part, but humans can be present at different locations in the scene. In order to make the Gaussian assumption reasonable, translations need to be removed from the positions. Therefore, we use a local coordinate system [33] for each triangle (A_t, B_t, C_t), i.e., we take one body part (for example A_t) as the origin, and use relative positions for other body parts. More formally, let $\overline{\mathbf{x}}$ denote a vector of positions $\overline{\mathbf{x}} = (x_{A_t}, x_{B_t}, x_{C_t}, y_{A_t}, y_{B_t}, y_{C_t})^T$, where x and y denote horizontal and vertical positions, respectively. Then, if we describe positions relative to A_t, we obtain $\overline{\mathbf{x}}' = (x_{B_t} - x_{A_t}, x_{C_t} - x_{A_t}, y_{B_t} - y_{A_t}, y_{C_t} - y_{A_t})^T$. This can be written as $\overline{\mathbf{x}}' = W\overline{\mathbf{x}}$, where

$$W = \begin{pmatrix} A & \underline{0} \\ \underline{0} & A \end{pmatrix}, \text{with } A = \begin{pmatrix} -1 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix}. \tag{8}$$

The above discussion is on the dimensions representing positions. For the dimensions representing velocities, translation does not need to be removed, and the corresponding "A" matrix (as in (8)) is an identity matrix. In the greedy search algorithm, the differential entropies of all the possible triplets are needed and different triplets have different origins. We first estimate the mean μ and covariance Σ of $\overline{X}^n$ (including all the body parts and without removing translation), then take the dimensions corresponding to the triangle and use equations

$$\mu' = \frac{1}{N}\sum_{n=1}^{N}\overline{\mathbf{x}}'^n = \frac{1}{N}\sum_{n=1}^{N}W\overline{\mathbf{x}}^n = W \cdot \frac{1}{N}\sum_{n=1}^{N}\overline{\mathbf{x}}^n = W\mu$$
$$\Sigma' = W\Sigma W^T$$

to get the translation invariant mean μ' and covariance Σ'. A similar procedure can be applied to pairs (for example, B_t can be taken as origin for (B_t, C_t)) to achieve translation invariance.

4 Unsupervised Learning of the Graph Structure

In this section, we present an algorithm to learn the probabilistic independence structure of human motion automatically from unlabeled training data. Our approach is based on maximizing the likelihood of the data. Taking the labeling (part assignments) as hidden variables, an EM-like algorithm can be applied. In the following, we first derive the algorithm assuming that all the foreground parts are observed for each training sample, and then generalize the algorithm to handle the case of missing body parts (occlusion).

4.1 Learning with All Foreground Parts Observed

This section develops an algorithm searching for the best decomposable triangulated model from unlabeled data, which is inspired by the idea of the expectation-maximization (or EM, [10], [34]) algorithm. The algorithm we propose does not guarantee the same convergence properties as EM although it works well in practice. Assume that we have a data set of N samples $\mathcal{X} = \{\overline{X}^1, \overline{X}^2, \ldots, \overline{X}^N\}$. Each sample $\overline{X}^n, 1 \le n \le N$, is a group of detected features containing the target object. Assume now that $\overline{X}^n$ is unlabeled, which means that the correspondence between the candidate features and the parts of the object is unknown.

For convenience, we first assume that all the foreground parts are observed for each sample. If the labeling for each $\overline{X}^n$ is taken as a hidden variable, then the idea of the EM algorithm can be used to learn the probability structure and parameters. Our method was inspired by [32], but while they assumed a jointly Gaussian probability density function, here we learn the probabilistic independence structure. Let h^n denote the labeling for $\overline{X}^n$. If $\overline{X}^n$ contains n_k features, then h^n is an n_k-dimensional vector with each element taking a value from $\mathcal{S}_{body} \cup \{BG\}$ ($\mathcal{S}_{body}$ is the set of body parts and BG is the background clutter label). The observations are $\mathcal{X} = \{\overline{X}^1, \overline{X}^2, \ldots, \overline{X}^N\}$, the hidden variables are $\mathcal{H} = \{h^n\}_{n=1}^N$, and the parameters to optimize are the probability (in)dependence structure and parameters for the associated probability density function. We use G to represent both the probability structure and the parameters. If we assume that $\overline{X}^n$ and $\overline{X}^m$ are independent when $n \ne m$ and h^n only depends on $\overline{X}^n$, then the likelihood function to maximize is,

$$\begin{aligned} L &= \log P(\mathcal{X}, G) = \sum\nolimits_{n=1}^{N} \log P(\overline{X}^n|G) + \log P(G) \\ &= \sum\nolimits_{n=1}^{N} \log \sum_{h_i^n \in H^n} P(\overline{X}^n, h^n = h_i^n|G) + \log P(G), \end{aligned} \tag{9}$$

where h_i^n is the ith possible labeling for $\overline{X}^n$, and H^n is the set of all such labelings. Optimization directly over (9) is hard, but it can be solved iteratively using the idea of EM. For each iteration t, we will optimize the function

$$\begin{aligned} Q(G_t|G_{t-1}) &= E[\log P(\mathcal{X}, \mathcal{H}, G_t)|\mathcal{X}, G_{t-1}] \\ &= \sum\nolimits_{n=1}^{N} E[\log P(\overline{X}^n, h^n, G_t)|\overline{X}^n, G_{t-1}] \\ &= \sum\nolimits_{n=1}^{N} \sum_{h_i^n \in H^n} P(h^n = h_i^n|\overline{X}^n, G_{t-1}) \\ &\quad \cdot \log P(\overline{X}^n, h^n = h_i^n, G_t) \\ &= \sum\nolimits_{n=1}^{N} \sum_{h_i^n \in H^n} R_i^n \log P(\overline{X}^n, h^n = h_i^n, G_t), \end{aligned} \tag{10}$$

where $R_i^n = P(h^n = h_i^n|\overline{X}^n, G_{t-1})$ is the probability of $h^n = h_i^n$ given the observation $\overline{X}^n$ and the decomposable probability structure G_{t-1}. R_i^n can be computed as,

$$R_i^n = P(h_i^n|\overline{X}^n, G_{t-1}) = P(\overline{X}^n, h_i^n, G_{t-1}) / \sum_{h_i^n} P(\overline{X}^n, h_i^n, G_{t-1}). \tag{11}$$

For each iteration t, R_i^n is a fixed number for a hypothesis h_i^n.

We use the same method as in Section 2.1 and [27], [26] to compute $P(\overline{X}^n, h_i^n, G)$ (G is G_t in (10) and G_{t-1} in (11)). Under the labeling hypothesis $h^n = h_i^n$, $\overline{X}^n$ is divided into the foreground features $\overline{X}_{fg}^n$, which are parts of the object, and background (clutter) $\overline{X}_{bg}^n$. If the foreground features $\overline{X}_{fg}^n$ are independent of clutter $\overline{X}_{bg}^n$, then,

$$\begin{aligned} P(\overline{X}^n, h_i^n, G) &= P(\overline{X}^n|h_i^n, G)P(h_i^n, G) \\ &= P(\overline{X}_{fg}^n|h_i^n, G)P(\overline{X}_{bg}^n|h_i^n, G)P(h_i^n|G)P(G). \end{aligned} \tag{12}$$

Substituting (12) into (10), we get,

$$\begin{aligned} &\sum_{n=1}^N \sum_{h_i^n \in H^n} R_i^n \log P(\overline{X}^n, h^n = h_i^n, G_t) \\ &= \sum_{n=1}^N \sum_{h_i^n \in H^n} R_i^n [\log P(\overline{X}_{fg}^n|h_i^n, G_t) + \log P(\overline{X}_{bg}^n|h_i^n, G_t) \\ &\quad + \log P(h_i^n|G_t) + \log P(G_t)] \\ &= \sum_n \sum_{h_i^n} R_i^n \log P(\overline{X}_{fg}^n|h_i^n, G_t) + \\ &\quad \sum_n \sum_{h_i^n} R_i^n \log P(\overline{X}_{bg}^n|h_i^n, G_t) + \\ &\quad \sum_n \sum_{h_i^n} R_i^n \log P(h_i^n|G_t) + \sum_n \sum_{h_i^n} R_i^n \log P(G_t). \end{aligned} \tag{13}$$

If we assume that the priors $P(h_i^n|G_t)$ are the same for different h_i^n, and $P(G_t)$ are the same for different decomposable triangulated graphs, the last two terms of (13) do not depend on G_t. If we assume independent uniform background noise[6] as in [32], [27], [26], then the second term $P(\overline{X}_{bg}^n|h_i^n, G_t) = (\frac{1}{S})^{n_k - M}$, where S is the volume of the space a background feature lies in and is not a function of G_t. Hence, we only need to optimize over the first term. Under probability decomposition G_t, $P(\overline{X}_{fg}^n|h_i^n, G_t)$ can be computed as in (2). Therefore, the maximization of (10) is equivalent to maximizing,

$$Q(G_t|G_{t-1}) \sim \sum_{n=1}^N \sum_{h_i^n} R_i^n \log[P(\overline{X}_{fg}^n|h_i^n, G_t)] \tag{14}$$

$$= \sum_{n=1}^N \sum_{h_i^n} R_i^n \left[\sum_{t=1}^T \log P(X_{A_t}^{ni}|X_{B_t}^{ni}, X_{C_t}^{ni}) + \log P(X_{B_T}^{ni}, X_{C_T}^{ni})\right], \tag{15}$$

where $X_{A_t}^{ni}$ is the measurements of body part A_t under labeling h_i^n for $\overline{X}^n$, etc. For most problems, the number of possible labelings is very large (on the order of $(n_k)^M$), and it is computationally prohibitive to sum over all the possible h_i^n as in (15). We take here the simplest approximation: if there is one hypothesis labeling h_i^{n*} that is much better than other hypotheses, i.e., R_i^{n*} corresponding to h_i^{n*} is much larger than other R_i^ns, then R_i^{n*} can be taken as 1 and other R_i^ns as 0. Hence, (15) can be approximated as

$$Q(G_t|G_{t-1}) \sim \sum_{n=1}^N \left[\sum_{t=1}^T \log P(X_{A_t}^{ni*}|X_{B_t}^{ni*}, X_{C_t}^{ni*}) + \log P(X_{B_T}^{ni*}, X_{C_T}^{ni*})\right], \tag{16}$$

where $X_{A_t}^{ni*}, X_{B_t}^{ni*}$, and $X_{C_t}^{ni*}$ are measurements corresponding to the best labeling h_i^{n*}, which can be obtained through the labeling algorithm presented in Section 2.1 using model G_{t-1}. Comparing (16) with (5), we know for iteration t, if the best hypothesis h_i^{n*} is used as the "true" labeling, then the decomposable triangulated graph structure G_t can be obtained through the algorithm described in Section 3. One approximation we make here is that the best hypothesis labeling h_i^{n*} for each $\overline{X}^n$ is really dominant among all the possible labelings so that hard assignment for labelings can be used. This is similar to the situation of K-means versus mixture of Gaussian for clustering problems [3]. Note that the best labeling is used to update the parameters of the probability density function (mean and covariance under Gaussian assumption). Therefore, in case of several labelings with close likelihoods, as long as the measurements associated with the body parts from these labelings are similar, the above approximation is still a good one.

The whole algorithm can be summarized as follows: given some random initial guess of the decomposable graph structure G_0 and its parameters, then for iteration t, (t is from 1 until the algorithm converges),

E-like step: Use G_{t-1} to find the best labeling h_i^{n*} for each $\overline{X}^n$. Let $\overline{X}_{fg}^{n*}$ denote the corresponding foreground measurements.

M-like step: update μ_t and covariance matrix Σ_t with $\mu_t = \frac{1}{N}\sum_n \overline{X}_{fg}^{n*}$ and $\Sigma_t = \frac{1}{N}\sum_n (\overline{X}_{fg}^{n*} - \mu_t)(\overline{X}_{fg}^{n*} - \mu_t)^T$. Use μ_t and Σ_t to compute differential entropies and run the graph growing algorithm described in Section 3 to get G_t.

Comparing with the standard EM technique, we made two approximations in the above procedure. In the E-like step, we use the best labeling instead of the weighted sum of all the possible labelings. Thus, our algorithm is clearly not EM, but rather another form of coordinate ascent. In the M-like step, there is no guarantee that the greedy graph growing algorithm

6. Uniform background noise is assumed mainly for computational simplicity. Uniform background noise is to assume that background features can be present anywhere with equal probability. For natural scenes, the independent assumption is not strictly true since closer features could be more correlated than far way features, but from an engineering point of view, our experience with experiments indicates that the assumption works fine.

will find the optimal graph. In Section 6, we evaluate these approximations with experiments on human motion.

4.2 Dealing with Missing Parts (Occlusion)

So far, we have assumed that all the parts are observed. When some parts are missing, the measurements for the missing body parts may be modeled as additional hidden variables [32] and the EM-like algorithm can be modified to handle the missing parts.

For each hypothesis labeling h^n, let $\overline{X}_o^n$ denote the measurements of the observed parts, $\overline{X}_m^n$ be the measurements for the missing parts, and $\overline{X}_{fg}^n = [\overline{X}_o^{nT} \overline{X}_m^{nT}]^T$ be the measurements of the whole object (to reduce clutter in the notation, we assume that the dimensions can be sorted in this way). The superscript T denotes transpose. For each iteration t, we need to compute μ_t and Σ_t to obtain the differential entropies and then G_t with its parameters. Taking h^n and $\overline{X}_m^n$ as hidden variables, we can get,

$$\mu_t = \frac{1}{N} \sum_n E(\overline{X}_{fg}^n) \tag{17}$$

$$\begin{aligned} \Sigma_t &= \frac{1}{N} \sum_n E(\overline{X}_{fg}^n - \mu_t)(\overline{X}_{fg}^n - \mu_t)^T \\ &= \frac{1}{N} \sum_n E(\overline{X}_{fg}^n \overline{X}_{fg}^{nT}) - \mu_t \mu_t^T, \end{aligned} \tag{18}$$

where $E(\overline{X}_{fg}^n) = [\overline{X}_o^{n*T} E(\overline{X}_m^{nT})]^T$, and

$$E(\overline{X}_{fg}^n \overline{X}_{fg}^{nT}) = \begin{bmatrix} \overline{X}_o^{n*} \overline{X}_o^{n*T} & \overline{X}_o^{n*} E(\overline{X}_m^{nT}) \\ E(\overline{X}_m^n) \overline{X}_o^{n*T} & E(\overline{X}_m^n \overline{X}_m^{nT}) \end{bmatrix}.$$

All the expectations $E(\cdot)$ are conditional expectations with respect to $\overline{X}^n, h^n = h_i^{n*}$ and decomposable graph structure G_{t-1}. Therefore, $\overline{X}_o^{n*}$ are the measurements of the observed foreground parts under $h^n = h_i^{n*}$. Since G_{t-1} is Gaussian distributed, conditional expectation $E(\overline{X}_m^n)$ and $E(\overline{X}_m^n \overline{X}_m^{nT})$ can be computed from observed parts $\overline{X}_o^{n*}$ and the mean and covariance matrix of G_{t-1}.

5 Mixtures of Decomposable Triangulated Models

5.1 Definition

In the previous sections, we model each triangle by a Gaussian distribution; therefore, the joint probability density function of all the parts is a Gaussian. To better express the variability and/or different phases of human motion, we extend the algorithms to mixtures of decomposable triangulated models, which are mixtures of Gaussian, with each component model being a Gaussian with conditional independence described by a decomposable triangulated graph. Each component model is relatively independent in the sense that different components can have different sets of body parts. Intuitively, a mixture model is a weighted sum of several individual decomposable triangulated models.

More formally, a C-component mixture model can be represented by $G = [G^1 G^2 \cdots G^C]$ and $\Pi = [\pi^1 \pi^2 \cdots \pi^C]$, where G^j, $j = 1, \ldots, C$ is a decomposable triangulated Gaussian model, and π^j is the prior probability of G^j. Each component model G^j has an independent set of body parts—some features corresponding to foreground body parts of one component model may be taken as background by another component model.

For an unlabeled observation $\overline{X}$, let c (taking a value from 1 to C) represent the random variable assigning a component model to $\overline{X}$, and h_j the random variable denoting the labeling of $\overline{X}$ under component model G^j. Since different component models may have different sets of body parts, a labeling must be associated with a particular component model. The probability of an unlabeled observation $\overline{X}$ is,

$$\begin{aligned} P(\overline{X}) &= \sum_{j=1}^{C} P(\overline{X}|c=j) P(c=j) \\ &= \sum_{j=1}^{C} \sum_{h_{ji} \in H_j} P(\overline{X}, h_j = h_{ji} | c=j) P(c=j), \end{aligned} \tag{19}$$

where h_{ji} is the ith possible labeling of $\overline{X}$ under component model j, and H_j is the set of all such possible labelings. In the above equation, $P(c=j) = \pi^j$ is the prior probability of component j and $P(\overline{X}, h_j = h_{ji}|c=j)$ can be computed in a similar way to (12).

5.2 Learning Rules

For clarity, we first assume that all the foreground parts are present for each component. Compared with the algorithm in Section 4.1, the observations are the same: $\mathcal{X} = \{\overline{X}^1, \overline{X}^2, \ldots, \overline{X}^N\}$. But, we have one more set of hidden variables $\mathcal{C} = \{c^n\}_{n=1}^N$, where c^n assigns a component (from 1 to C) to $\overline{X}^n$, and $\mathcal{H}$, the set of random variables for labeling, becomes $\mathcal{H} = \{h^n\}_{n=1}^N$, where $h^n = \{h_j^n\}_{j=1}^C$, and h_j^n is the labeling of $\overline{X}^n$ under the jth component model. The parameters to estimate are the multiple components model G and the prior probabilities Π. By Bayes' rule and (19), the likelihood function we want to maximize is

$$\begin{aligned} L &= \log P(\mathcal{X}, G, \Pi) = \sum_{n=1}^{N} \log P(\overline{X}^n | G, \Pi) + \log P(G, \Pi) \\ &= \sum_{n=1}^{N} \log \sum_{j=1}^{C} \sum_{h_{ji}^n \in H_j^n} P(\overline{X}^n, h_j^n = h_{ji}^n, c^n = j | G, \Pi) \\ &\quad + \log P(G, \Pi), \end{aligned} \tag{20}$$

where h^n_{ji} is the ith possible labeling of $\overline{X}^n$ under the jth component model, and H^n_j is the set of all such possible labelings. Optimization directly over (20) is hard, and it can be solved iteratively. Let $G_t = [G^1_t G^2_t \cdots G^C_t]$ and $\Pi_t = [\pi^1_t \pi^2_t \cdots \pi^C_t]$ denote the parameters at iteration t. Then, at each iteration t, we will optimize the function

$$
\begin{aligned}
&Q(G_t, \Pi_t | G_{t-1}, \Pi_{t-1}) \\
&= E[\log P(\mathcal{X}, \mathcal{H}, \mathcal{C}, G_t, \Pi_t) | \mathcal{X}, G_{t-1}, \Pi_{t-1}] && (21)\\
&= \sum_{n=1}^{N} E[\log P(\overline{X}^n, h^n, c^n, G_t, \Pi_t) | \overline{X}^n, G_{t-1}, \Pi_{t-1}] && (22)\\
&= \sum_{n=1}^{N} \sum_{j=1}^{C} \sum_{h^n_{ji} \in H^n_j} P(h^n_j = h^n_{ji}, c^n = j | \overline{X}^n, G_{t-1}, \Pi_{t-1}) \\
&\quad \cdot \log P(\overline{X}^n, h^n_j = h^n_{ji}, c^n = j, G_t, \Pi_t) && (23)\\
&= \sum_{n=1}^{N} \sum_{j=1}^{C} \sum_{h^n_{ji} \in H^n_j} P(h^n_j = h^n_{ji} | c^n = j, \overline{X}^n, G_{t-1}, \Pi_{t-1}) \\
&\quad \cdot P(c^n = j | \overline{X}^n, G_{t-1}, \Pi_{t-1}) \\
&\quad \cdot \log P(\overline{X}^n, h^n_j = h^n_{ji}, c^n = j, G_t, \Pi_t). && (24)
\end{aligned}
$$

The $E[\cdot]$ in (21) and (22) is the expectation of log likelihood given the observed data and parameters from iteration $t-1$. Equation (23) is to compute the expectation by summing over all the possible values of the hidden variables. For convenience, we define $R^n_{ji} = P(h^n_j = h^n_{ji} | c^n = j,\ \overline{X}^n, G_{t-1}, \Pi_{t-1})$, which is the probability of a labeling h^n_{ji} of $\overline{X}^n$ given $\overline{X}^n$ and $\overline{X}^n$ belonging to component j, and $\omega^n_j = P(c^n = j | \overline{X}^n, G_{t-1}, \Pi_{t-1})$, which is the probability of $\overline{X}^n$ belonging to component j given $\overline{X}^n$. Assuming that all the component models have the same number of body parts and within one component model the prior probabilities of all the possible labelings are uniformly distributed, we can obtain (see [28] for detailed derivation),

$$
\omega^n_j = \frac{\pi^j_{t-1} \sum_{h^n_{ji} \in H^n_j} P(\overline{X}^n_{fg(ji)} | h^n_{ji}, G^j_{t-1})}{\sum_{k=1}^{C} \pi^k_{t-1} \sum_{h^n_{ki} \in H^n_k} P(\overline{X}^n_{fg(ki)} | h^n_{ki}, G^k_{t-1})}, \qquad (25)
$$

where $\overline{X}^n_{fg(ki)}$, $k = 1, \ldots, C$ is the foreground measurements of labeling $h^n_{ki} \in H^n_k$ under component model k. Since each G^k_{t-1}, $k = 1, \ldots, C$ is a decomposable triangulated Gaussian model, the summation $\sum_{h^n_{ki} \in H^n_k} P(\overline{X}^n_{fg(ki)} | h^n_{ki}, G^k_{t-1})$ in (25) can be computed efficiently by dynamic programming (use "sum" operation instead of "max" operation, for more details see [26]).

The computation of R^n_{ji} is the same as (11), but using component model G^j_{t-1}. ω^n_j and R^n_{ji} are computed using the parameters from iteration $t-1$, hence they are fixed constants for function Q at iteration t.

Substituting ω^n_j and R^n_{ji} into (24), we get,

$$
\begin{aligned}
&Q(G_t, \Pi_t | G_{t-1}, \Pi_{t-1}) \\
&= \sum_{n=1}^{N} \sum_{j=1}^{C} \sum_{h^n_{ji} \in H^n_j} R^n_{ji} \cdot \omega^n_j \cdot \log P(\overline{X}^n, h^n_j = h^n_{ji}, c^n = j, G_t, \Pi_t) \\
&= \sum_{n=1}^{N} \sum_{j=1}^{C} \omega^n_j \sum_{h^n_{ji} \in H^n_j} R^n_{ji} \cdot \log P(\overline{X}^n, h^n_j = h^n_{ji}, c^n = j, G_t, \Pi_t) \\
&= \sum_{n=1}^{N} \sum_{j=1}^{C} \omega^n_j \sum_{h^n_{ji} \in H^n_j} R^n_{ji} \cdot [\log P(\overline{X}^n | h^n_j = h^n_{ji}, c^n = j, G_t, \Pi_t) \\
&\quad + \log P(h^n_j = h^n_{ji} | c^n = j, G_t, \Pi_t) + \log P(c^n = j | G_t, \Pi_t)] \\
&= \sum_{n=1}^{N} \sum_{j=1}^{C} \omega^n_j \sum_{h^n_{ji} \in H^n_j} R^n_{ji} \cdot [\log P(\overline{X}^n_{fg(ji)} | h^n_j = h^n_{ji}, c^n = j, G_t, \Pi_t) \\
&\quad + \log P(\overline{X}^n_{bg(ji)} | h^n_j = h^n_{ji}, c^n = j, G_t, \Pi_t) \\
&\quad + \log P(h^n_j = h^n_{ji} | c^n = j, G_t, \Pi_t) + \log P(c^n = j | G_t, \Pi_t)] \\
&= Q_1 + Q_2 + Q_3 + Q_4, && (26)
\end{aligned}
$$

where

$$
\begin{aligned}
Q_1 &= \sum_{n=1}^{N} \sum_{j=1}^{C} \omega^n_j \sum_{h^n_{ji} \in H^n_j} R^n_{ji} \cdot \log P(\overline{X}^n_{fg(ji)} | h^n_j = h^n_{ji}, c^n = j, G_t, \Pi_t) \\
&= \sum_{j=1}^{C} \sum_{n=1}^{N} \omega^n_j \sum_{h^n_{ji} \in H^n_j} R^n_{ji} \cdot \log P(\overline{X}^n_{fg(ji)} | h^n_j = h^n_{ji}, G^j_t) \\
&= \sum_{j=1}^{C} Q^j_1 && (27)\\
Q_2 &= \sum_{n=1}^{N} \sum_{j=1}^{C} \omega^n_j \sum_{h^n_{ji} \in H^n_j} R^n_{ji} \\
&\quad \cdot \log P(\overline{X}^n_{bg(ji)} | h^n_j = h^n_{ji}, c^n = j, G_t, \Pi_t) && (28)\\
Q_3 &= \sum_{n=1}^{N} \sum_{j=1}^{C} \omega^n_j \sum_{h^n_{ji} \in H^n_j} R^n_{ji} \cdot \log P(h^n_j = h^n_{ji} | c^n = j, G_t, \Pi_t) && (29)\\
Q_4 &= \sum_{n=1}^{N} \sum_{j=1}^{C} \omega^n_j \sum_{h^n_{ji} \in H^n_j} R^n_{ji} \cdot \log P(c^n = j | G_t, \Pi_t) \\
&= \sum_{j=1}^{C} \sum_{n=1}^{N} \omega^n_j \sum_{h^n_{ji} \in H^n_j} R^n_{ji} \cdot \log(\pi^j_t) \\
&= \sum_{j=1}^{C} \left(\sum_{n=1}^{N} \omega^n_j \right) \log(\pi^j_t). && (30)
\end{aligned}
$$

We want to find G_t and Π_t which can maximize $Q = Q_1 + Q_2 + Q_3 + Q_4$. Q_2 and Q_3 are not functions of G_t and Π_t. Q_1 is a function of G_t and Q_4 is a function of Π_t. From (27), the best G^j_t is the one which can maximize

$$
\begin{aligned}
Q^j_1 &= \sum_{n=1}^{N} \omega^n_j \sum_{h^n_{ji} \in H^n_j} R^n_{ji} \cdot \log P(\overline{X}^n_{fg(ji)} | h^n_j = h^n_{ji}, G^j_t) && (31)\\
&\approx \sum_{n=1}^{N} \omega^n_j \log P(\overline{X}^{n*}_{fg(ji)} | G^j_t), && (32)
\end{aligned}
$$

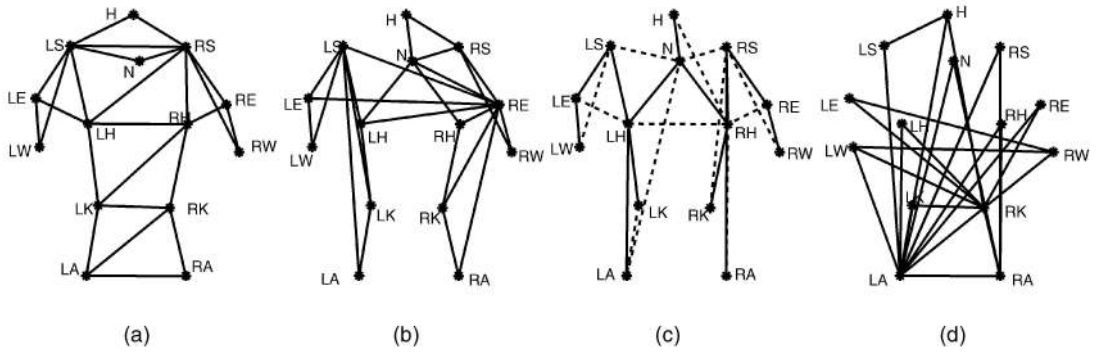


Fig. 2. Decomposable triangulated models for motion capture data. (a) Hand-constructed model [27]. (b) Model obtained from greedy search (Section 3.2). (c) Decomposable triangulated model grown from a maximum spanning tree [7], [22], [17]. The solid lines are edges from the maximum spanning tree and the dashed lines are added edges [28]. (d) A randomly generated decomposable triangulated model.

where $\overline{X}^{n*}_{fg(ji)}$ is the foreground configuration with the highest R^n_{ji}, i.e., the best labeling of $\overline{X}^n$ under model G^j_{t-1}. The approximation from (31) to (32) is under the same reasoning as from (15) to (16). Under Gaussian assumption, the maximum likelihood parameter estimation of G^j_t can be obtained by taking derivatives of (32) with respect to mean and covariance matrix and equating to zero. Then, we have the updated parameters,

$$\mu^j_t = \frac{\sum_n \omega^n_j \overline{X}^{n*}_{fg(ji)}}{\sum_n \omega^n_j}, \tag{33}$$

$$\Sigma^j_t = \frac{\sum_n \omega^n_j \overline{X}^{n*}_{fg(ji)} (\overline{X}^{n*}_{fg(ji)})^T}{\sum_n \omega^n_j} - \mu^j_t (\mu^j_t)^T. \tag{34}$$

From μ^j_t and Σ^j_t, the decomposable triangulated structure can be obtained by running the graph growing algorithm in Section 3.

To optimize Π_t, we maximize Q_4 under the constraint $\sum_{j=1}^C \pi^j_t = 1$. Using Lagrange multipliers, we get,

$$\pi^j_t = \frac{\sum_n \omega^n_j}{N}. \tag{35}$$

The whole algorithm can be summarized as follows: First, we need to fix C, the number of component models in the mixtures, and the number of body parts in each component model. Then we generate random initializations for each component model, $G_0 = [G^1_0, \ldots, G^C_0]$, and the initial priors Π_0. At each iteration t (t from 1 till convergence),

E-like step: For each $\overline{X}^n$, find the best labeling $\overline{X}^{n*}_{fg(ji)}$ using component model G^j_{t-1}, $j = 1, \ldots, C$ and compute ω^n_j by (25).

M-like step: Update μ^j_t and Σ^j_t as in (33) and (34). Run the graph growing algorithm (Section 3) on each Σ^j_t to obtain updated G^j_t, $j = 1, \ldots, C$. Update Π_t as in (35).

So far, we have assumed that all the foreground parts are observed for each component model. In the case of some parts missing (occlusion), the same techniques as in Section 4.2 are applied.

5.3 Detection and Labeling Using Mixture Models

For an observation $\overline{X}$, we can run the detection and labelings algorithms summarized in Section 2.1 on each component model G^j, $j = 1, \ldots, C$, to get the best labeling $\overline{X}^*_{fg(j)}$ and an estimation of $P_{G^j}(\overline{X})$ (by either the winner-take-all strategy or the sum-over-all-possible-labeling strategy). Detection can be performed by thresholding $\sum_{j=1}^C \pi^j \cdot P_{G^j}(\overline{X})$. The localization of the human body can be determined by the best configuration $\overline{X}^*_{fg(j)}$ with the highest $\pi^j \cdot P_{G^j}(\overline{X})$ among all the best configurations $\overline{X}^*_{fg(j)}$, $j = 1, \ldots, C$.

6 Experiments

Experiments were performed on two types of data: labeled motion capture data and unlabeled real image sequences. The experiments on the labeled motion capture data were used to test the greedy graph growing algorithm described in Section 3. The experiments on the unlabeled real image sequences were to evaluate the unsupervised learning algorithms developed in Sections 4 and 5.

6.1 Experiments on Motion Capture Data

Our motion capture data (the same as in [27]) consist of the 3D positions of 14 markers fixed rigidly on a subject's body. These positions were tracked with 1mm accuracy as the subject walked back and forth, and projected to 2D. We used around 3,000 frames (50 seconds long) to build models, and another 3,000 frames for testing.

Under the Gaussian assumption, we first estimated the joint probability density function (mean and covariance) of the data. From the estimated mean and covariance, we could compute differential entropies for all the possible triplets and pairs and further run the greedy search algorithm (Section 3.2) to find the approximate best triangulated model. In order to benchmark the algorithm, we also obtained a maximum spanning tree (MST) based on differential entropies [7], [22], [17], and edges were added in a greedy fashion to transform the MST into a decomposable triangulated graph [28]. Fig. 2 displays the models. Fig. 2a is the hand-constructed model used in previous work [27] (Fig. 1a); Fig. 2b is the model obtained from greedy search (Section 3.2); Fig. 2c is the decomposable triangulated model grown from a maximum

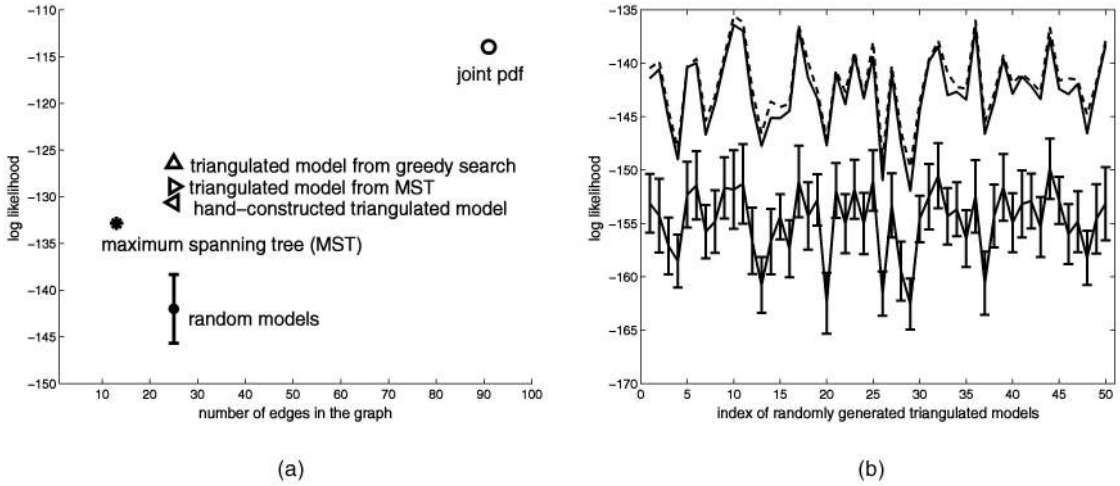


Fig. 3. Likelihood evaluation of graph growing algorithms. (a) On motion capture data. Log likelihood versus number of edges in the model. (b) On synthetic data with decomposable triangulated independence. Dashed curve: likelihoods of the true models, solid curve: of models from greedy search, and the solid line with error bars: of random triangulated models.

Fig. 4. Sample images extracted from our sequences. The text string in parenthesis indicates the image type.

spanning tree. The solid lines are edges of the maximum spanning tree and the dashed lines are added edges. Fig. 2d shows a randomly generated decomposable triangulated model, which is grown as follows: we start from a randomly selected edge. At each following stage, a vertex is randomly selected and an edge in the existing graph is randomly

TABLE 1
Types of Images Used in the Experiments

code-name	description
p1	person walking R-L. 10 subjects. Subject LG (12 x 80); other subjects (3-4 x 80) each.
p2	person walking R-L with another person biking either R-L or L-R. (4 x 50)
p3	person walking R-L with a car driving R-L. (4 x 40-60)
b+	person biking R-L alone or with another person walking L-R. (3 x 40)
b-	person biking L-R alone or with another person walking L-R. (5 x 40)
c+	car moving R-L. (2 x 70)
c-	car moving L-R alone (1 x 100) or car moving L-R with a person walking L-R (1 x 50)
r+	person running R-L. (6 x 30)
r-	person running L-R. (6 x 30)
w+	water running R-L. (1 x 30)
cp+	stationary background (no person) with camera panning L-R. (1 x 50)
cp-	stationary background (no person) with camera panning R-L. (1 x 50)
cps+	stationary scene (with person standing still) with camera panning L-R. (2 x 50)
cps-	stationary scene (with person standing still) with camera panning R-L. (2 x 50)
cpt+	person walking L-R and camera panning L-R to follow the person. (2 x 50)
cpt-	person walking R-L and camera panning R-L to follow the person. (2 x 50)

"L-R" denotes "from left to right," and "R-L" means "from right to left." The digits in the parenthesis are the number of sequences by the number of frames in each sequence. For example, (3-4 x 80) means that there are three or four sequences, with around 80 frames for each sequence. The +/- in the code-names denotes whether movement is R-L or L-R.

selected as its parent edge and then the newly selected vertex is connected with the two vertices of the edge.

Since the goal of model searching is to find the one with the highest likelihood (Section 3.1), we first evaluate the models by likelihoods. Fig. 3a shows the likelihood of the estimated joint probability density function (pdf), for each one of the approximate models as well as a number of randomly generated models (mean and error bars). The horizontal axis is the number of edges in the model, which is an indicator of computational cost. The decomposable triangulated model from the greedy search (Section 3.2) has the highest likelihood of all the approximate models. The triangulated model grown from maximum spanning tree is the second best. The hand-constructed model is the third best. The maximum spanning tree is worse than the above three triangulated models (not surprisingly, since it has fewer parameters), but is superior to the random triangulated models. The full Gaussian joint pdf shown for comparison has the highest likelihood, but it cannot be used in a computationally efficient manner.

A natural question to ask is: how close is the likelihood of our greedy graph to the likelihood of the "optimal" triangulated graph? We address this question with experiments on synthetic datasets generated by models with known decomposable triangulated independence. To accomplish this, we generate a random decomposable triangulated model, then generate data according to this model: 3,000 frames for learning and 3,000 frames for testing. In order to make this a meaningful comparison, we add the constraint that, on each triangle, the marginal probability density of the generated data is the same as that of the original motion capture data. Fig. 3b shows likelihoods using 50 synthetic data sets, which were generated from 50 triangulated models. The likelihood of the greedy algorithm (solid curve) matches the likelihood of the true model (dashed curve) very well. The solid line with error

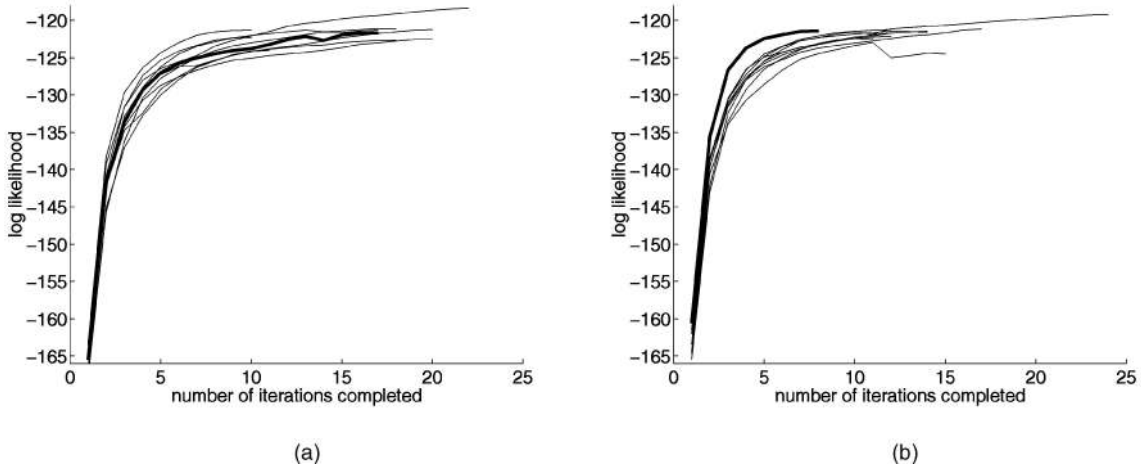


Fig. 5. Evaluation of the unsupervised learning algorithm: evolution of log-likelihoods from different random initializations. The indices along the horizontal axis show the number of iterations completed. (a) Shows 12-part 3-component single-subject models. (b) Shows 12-part 3-component multiple-people models.

bars are the likelihoods of random triangulated models. We conclude that the greedy search algorithm (Section 3.2) delivers quasioptimal solutions on this type of data. We will therefore use this algorithm in the following experiments.

In this section, we used a criterion based on likelihood to evaluate the greedy graph growing algorithm. However, there are other more important factors such as ability of dealing with occlusion and translation invariance that make decomposable triangulated graphs an appropriate choice for our application.

6.2 Experiments on Gray-Scale Image Sequences

In this section, we conduct experiments on gray-scale image sequences. The image sequences were acquired using a digital camcorder at 30Hz frame rate. The images were converted into gray-scale and the image resolution is 240 x 360. To apply our algorithms, candidate features were obtained using a Lucas-Tomasi-Kanade [30] feature selector/tracker on pairs of frames. Features are selected at each frame, and are tracked to the next frame to obtain positions and velocities [26].

The image sequences (see Figs. 4 and 8 for sample images) used in the experiments are summarized in Table 1. The first column of Table 1 gives the code-names of the sequences, e.g., (p1), (p2), and (b-), which will be used to represent the sequences. In the description of the sequences, "L-R" denotes "from left to right," and "R-L" means "from right to left." The 10 subjects of the (p1) sequences include six males and four females from 20 to 50 years old. We assume that the distance between the person and the camera is constant.

In the experiments, R-L walking motion models were learned from (p1) sequences and tested on all types of sequences to see if the learned model can detect R-L walking and label the body parts correctly. Type (p1), (p2), and (p3) sequences are considered as positive examples and the others are negative examples. In the following, we first evaluate the learning algorithms and then report the detection and labeling results.

6.2.1 Evaluation of the Unsupervised Learning Algorithm

There are two approximations in the unsupervised learning algorithms (see the end of Section 4.1). Here, we evaluate the algorithm by checking how the log-likelihoods evolve with each iteration and if they converge. We learn two types of models. The first one is a single-subject model: using nine type (p1) sequences of one subject named LG. The other is a multiple-people model learned from 12 type (p1) sequences from four subjects (including subject LG).

Fig. 5 shows some typical results of learning a 3-component model, each component with 12 parts. Fig. 5a is of single-subject models and Fig. 5b is of multiple-people models. We used random initializations and the 10 curves in Fig. 5a or Fig. 5b correspond to 10 such random initializations. If the likelihood difference of two iterations is less than 0.1 percent, the algorithm terminates. From Fig. 5, we can see that while log likelihood is not strictly monotonic, but, in general, the log likelihoods increase with each iteration and converge well.

6.2.2 Models Obtained

We tested the models using a small validation set and found no big difference in terms of detection performance. Figs. 6a and 6b show a single-subject model (corresponding to the thick curve in Fig. 5a). Fig. 6a gives the mean positions and mean velocities (shown in arrows) of the parts for each component model. The prior probabilities are shown on top of each plot. Fig. 6b depicts the learned decomposable triangulated probabilistic structure for the three component models in Fig. 6a, respectively. The letter labels show the body parts correspondence. Figs. 6c and 6d are a multiple-people model (corresponding to the thick curve in Fig. 5b) and follow the same representation custom as in Figs. 6a and 6b.

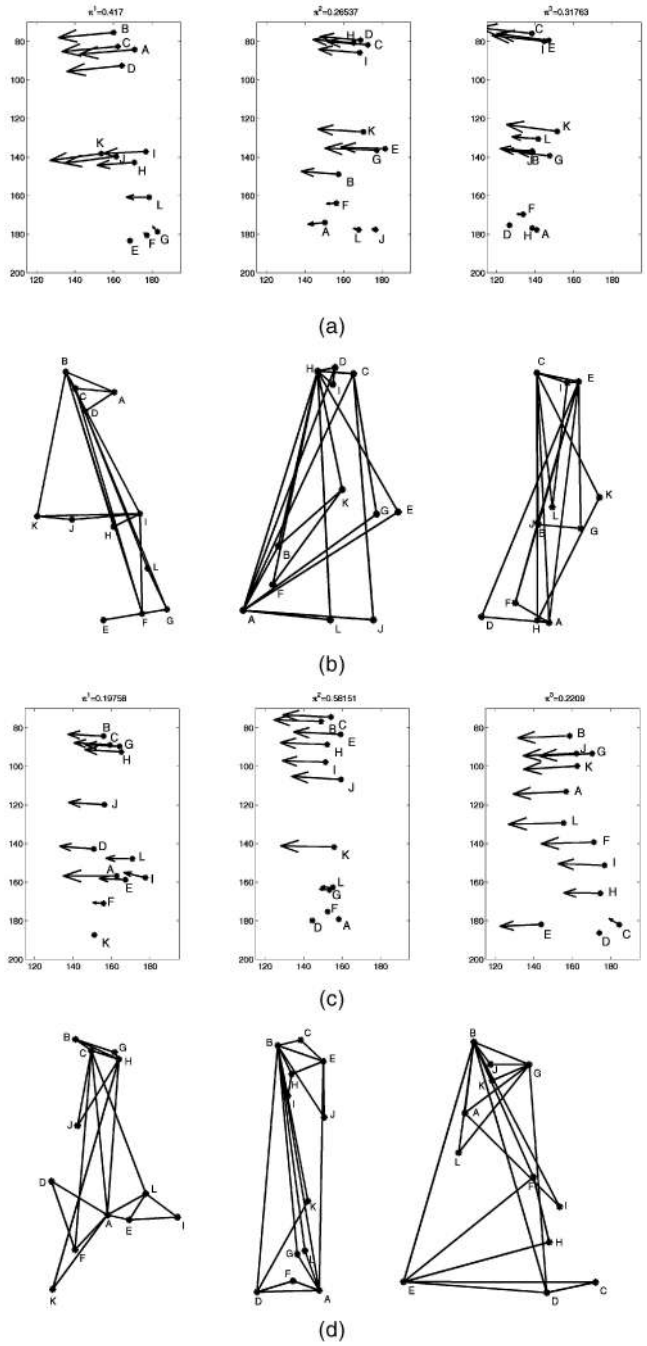
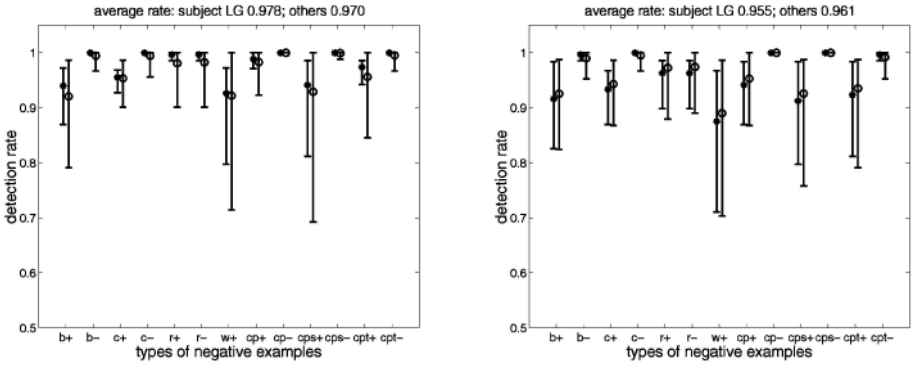


Fig. 6. Examples of 12-part 3-component models. (a) and (b) are a single-subject model (corresponding to the thick curve in Fig. 5a), and (c) and (d) are a multiple-people model (the thick curve in Fig. 5b). (a) (or (c)) gives the mean positions and mean velocities (shown in arrows) of the parts for each component model. The number π_i, $i = 1, 2, 3$, on top of each plot is the prior probability for each component model. (b) (or (d)) is the learned decomposable triangulated probabilistic structure for models in (a) (or (c)). The letter labels show the body parts correspondence.



(a) (b)

Fig. 7. Detection rates versus types of negative examples. (a) is from the single-subject model (Fig. 6a) and (b) is from the multiple-people model (Fig. 6c). Stars (*) with error bars use R-L walking sequences of subject LG as positive examples and circles (o) with error bars use R-L walking sequences of other subjects. The stars (or circles) show the average detection rates and error bars give the maximum and minimum detection rates.

6.2.3 Detection and Labeling

We conduct detection and labeling (Section 5.3) experiments using the models obtained. To quantify the detection performance, we first get receiver operating characteristics (ROC) curves from the likelihood of each sample (Section 5.3). From an ROC curve, we can take the "equal error" detection rate when $P_{detection} = 1 - P_{false\ alarm}$ as an indicator of the detection performance. The performance is measured on each pair of frames independently. Fig. 7 summarizes such detection rates of positive R-L walking sequences versus different types of negative sequences. The horizontal axis of Fig. 7 displays the different types of negative examples (as described in Table 1). We get the detection rate of each positive R-L walking sequence versus a certain type of negative sequences, and the average detection rate is shown either in star (*) or in circle (o). The error bars show the maximum or minimum detection rate. The stars (*) with error bars use the positive walking sequences of subject LG as positive examples, and the circles (o) with error bars use the positive sequences of other subjects not in the training set. Fig. 7a is from the single-subject model as in Fig. 6a, and Fig. 7b is from the multiple-people model as in Fig. 6c.

All the negative sequences ending with (+) have R-L motion and (-) means that L-R motion is the major motion. Detection is almost perfect when images from an L-R (-) type of sequences are used as negative examples. Among the R-L (+) types of sequences, the water moving R-L sequence (with a lot of features) and the sequences of a person standing still with camera panning are the hardest. From Fig. 7, we see that the two models perform similarly, with overall detection rates (out-of-training-set subjects) of 97.0 percent and 96.1 percent for the single-subject model and multiple-people model, respectively.

We also experimented with a 12-part Gaussian model with a single component. We find that the detection rates are similar to the 3-component models when the negative sequences offer easy discrimination, e.g., L-R (-), but the detection rates are approximately 10 percent worse than 3-component models on hard discrimination tasks, e.g., water running R-L (w+) sequences.

Fig. 8 shows results on some images using the 12-part 3-component multiple-people model (Fig. 6c). The text string at the bottom right corner of each image indicates which type of sequences the image is from. The small black circles are candidate features obtained from the Lucas-Tomasi-Kanade feature detector/tracker. The arrows associated with circles indicate the velocities. The horizontal lines at the bottom left of each image give the log-likelihoods. The top three lines are the log-likelihoods ($P_{G^j}(\overline{X})$) of the three component models, respectively. The bottom line is the overall log-likelihood ($\sum_{j=1}^{C} \pi^j \cdot P_{G^j}(\overline{X})$) (Section 5.3). The short vertical bar (at the bottom) indicates the threshold for detection, under which we get equal missed detection rate and false alarm rate for all the available positive and negative examples. If a R-L walking motion is detected according to the threshold, then the best labeling from the component with the highest log-likelihood is drawn in solid black dots, and the letter beside each dot shows the correspondence with the parts of the component model in Fig. 6c. The number at the upper right corner shows the highest likelihood component, with $1, 2, 3$ corresponding to the three components in Fig. 6c from left to right. For the samples in Fig. 8, all the positive R-L walking are correctly detected, and one negative example (from the water running R-L sequence) is wrongly claimed as a person R-L walking (a false alarm).

7 Conclusions and Discussions

We have described a method for learning a probabilistic model of human motion in an unsupervised fashion from unlabeled cluttered data. Our models are mixtures of Gaussian with conditional independence described by a decomposable triangulated graph. We explore the efficiency and effectiveness of our algorithm by learning a model of right-to-left walking and testing on walking sequences of a number of people as well as a variety of nonwalking motions. We find an average of 4 percent error rate on our examples. This rate is measured on pairs of frames evaluated independently, and it becomes virtually zero when 4-5 pairs of frames (150-200 ms of video) are considered simultaneously [27], [26]. This is very promising for building a real-life system, for example, a pedestrian detector.

We find that our models generalize well across subjects and not at all across types of motions. The model learned on

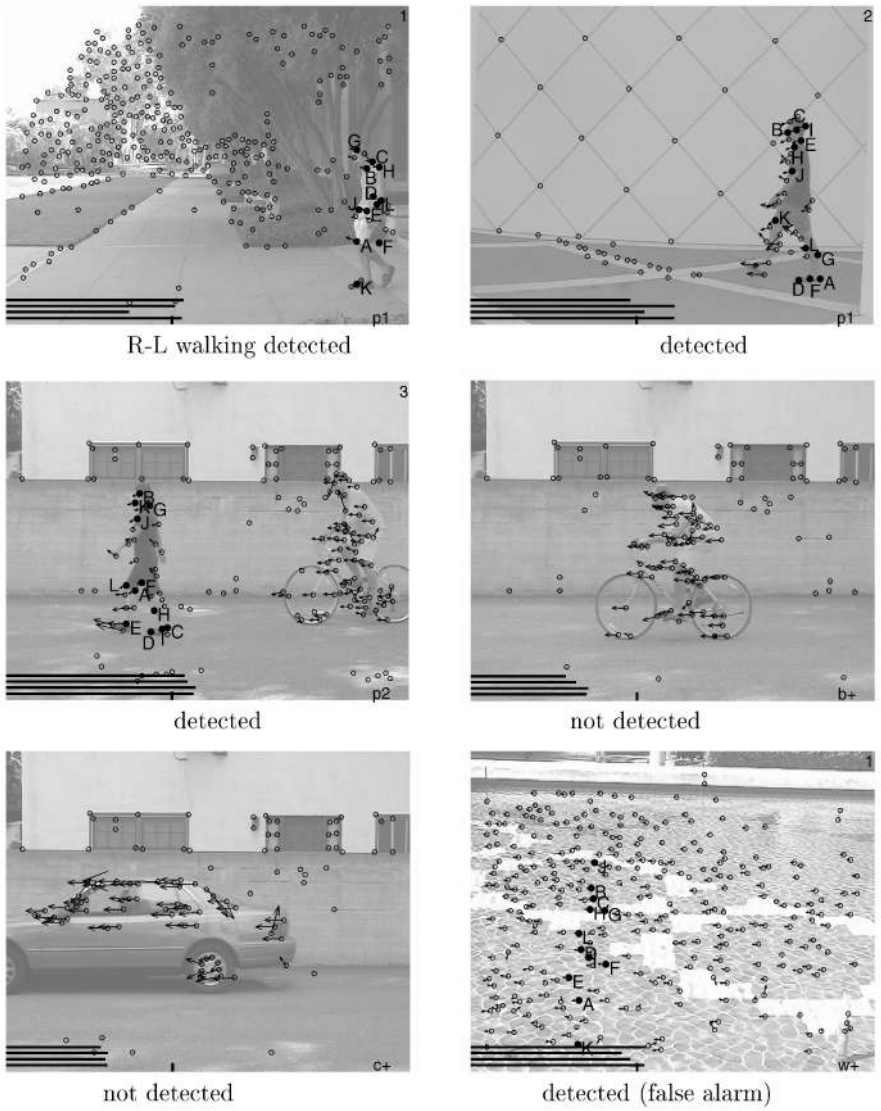


Fig. 8. Detection and labeling results on some images. If a R-L walking motion is detected according to the threshold, then the best labeling from the component with the highest log-likelihood is drawn in solid black dots. The number at the upper right corner shows the highest likelihood component. See text for detailed explanations of symbols.

subject LG worked equally well in detecting all other subjects and very poorly at subject discrimination. By contrast, it was easy to discriminate walking from jogging and biking in the same direction.

We used point features (from a corner detector) in our experiments because they are easier to obtain compared to body segments that may be hard to detect in case of severe occlusion. Another reason is that psychophysics experiments (Johansson's experiments [19]) show that the human visual system can perceive vivid human motion from moving dots representing the motion of the main joints of the human body. But, the algorithms can also be applied to other types of features. A systematic study of the trade-off between model complexity (number of components and number of parts) and accuracy is still missing (we used 3-component 12-part models in this paper), as well as experiments with different types of motions beyond walking. Decomposable triangulated graphs are used in our application because intuitively they have better graph connectivity in case of occlusion and, therefore, better ability in achieving translation invariance (Sections 2.2 and 6.1). However, while trees appear less promising than triangulated graphs, we have not yet carried out experiments to confirm our intuition. Since the unsupervised technique described in this paper is not limited to decomposable triangulated graphs, it would be equally interesting to experiment with other types of graphical models.

ACKNOWLEDGMENTS

Part of the work in this paper was published in the Proceedings of IEEE Conference on Computer Vision and Pattern Recognition '01 and NIPS '01. This work was funded by the US National Science Foundation Engineering Research Center for Neuromorphic Systems Engineering (CNSE) at Caltech (NSF9402726), and an Office of Navy Research grant N00014-01-1-0890 under the MURI program. The authors would like to thank Charless Fowlkes for bringing the Chow and Liu paper to their attention.

REFERENCES

[1] Y. Amit and A. Kong, "Graphical Templates for Model Registration," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 18, pp. 225-236, 1996.

[2] U. Bertele and F. Brioschi, *Nonserial Dynamic Programming.* Academic Press, 1971.

[3] C.M. Bishop, *Neural Networks for Pattern Recognition.* Clarendon Press, 1995.

[4] A. Blake and M. Isard, "3D Position, Attitude and Shape Input Using Video Tracking of Hands and Lips," *Proc. ACM Siggraph,* pp. 185-192, 1994.

[5] C. Bregler and J. Malik, "Tracking People with Twists and Exponential Maps," *Proc. IEEE Computer Vision and Pattern Recognition,* pp. 8-15, 1998.

[6] D. Chickering, D. Geiger, and D. Heckerman, "Learning Bayesian Networks Is NP-Hard," technical report, Microsoft Research, MSR-TR-94-17, 1994.

[7] C.K. Chow and C.N. Liu, "Approximating Discrete Probability Distributions with Dependence Trees," *IEEE Trans. Information Theory,* vol. 14, pp. 462-467, 1968.

[8] T.H. Cormen, C.E. Leiserson, and R.L. Rivest, *An Introduction to Algorithms.* MIT Press, 1990.

[9] T.M. Cover and J.A. Thomas, *Elements of Information Theory.* John Wiley and Sons, 1991.

[10] A. Dempster, N. Laird, and D. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *J. Royal Statistical Soc. B,* vol. 39, pp. 1-38, 1977.

[11] C. Fowlkes, "Labeling Human Motion Using Mixtures of Trees," Univ. of California at Berkeley, personal communication, 2001.

[12] N. Friedman and M. Goldszmidt, "Learning Bayesian Networks from Data," AAAI 1998 Tutorial, http://robotics.stanford.edu/people/nir/tutorial/, 1998.

[13] N. Friedman and D. Koller, "Being Bayesian about Network Structure," *Proc. 16th Conf. Uncertainty in Artificial Intelligence,* pp. 201-210, 2000.

[14] D. Gavrila, "The Visual Analysis of Human Movement: A Survey," *Computer Vision and Image Understanding,* vol. 73, pp. 82-98, 1999.

[15] L. Goncalves, E. Di Bernardo, E. Ursella, and P. Perona, "Monocular Tracking of the Human Arm in 3D," *Proc. Fifth Int'l Conf. Computer Vision,* pp. 764-770, June 1995.

[16] I. Haritaoglu, D. Harwood, and L. Davis, "Who, When, Where, What: A Real Time System for Detecting and Tracking People," *Proc. Third Face and Gesture Recognition Conf.,* pp. 222-227, 1998.

[17] S. Ioffe and D. Forsyth, "Human Tracking with Mixtures of Trees," *Proc. Int'l Conf. Computer Vision,* pp. 690-695, July 2001.

[18] F.V. Jensen, *An Introduction to Bayesian Networks.* Springer, 1996.

[19] G. Johansson, "Visual Perception of Biological Motion and a Model for Its Analysis," *Perception and Psychophysics,* vol. 14, pp. 201-211, 1973.

[20] *Learning in Graphical Models,* M.I. Jordan, ed. MIT Press, 1999.

[21] M.I. Jordan, *An Introduction to Graphical Models.* to be published.

[22] M. Meila and M. I. Jordan, "Learning with Mixtures of Trees," *J. Machine Learning Research,* vol. 1, pp. 1-48, 2000.

[23] R. Polana and R.C. Nelson, "Detecting Activities," *Proc. DARPA Image Understanding Workshop,* pp. 569-574, 1993.

[24] J.M. Rehg and T. Kanade, "Visual Tracking of High DOF Articulated Structures: An Application to Human Hand Tracking," *Proc. European Conf. Computer Vision,* vol. 2, pp. 35-46, 1994.

[25] K. Rohr, "Incremental Recognition of Pedestrians from Image Sequences," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 8-13, June 1993.

[26] Y. Song, X. Feng, and P. Perona, "Towards Detection of Human Motion," *Proc. IEEE Computer Vision and Pattern Recognition,* vol. 1, pp. 810-817, June 2000.

[27] Y. Song, L. Goncalves, E. Di Bernardo, and P. Perona, "Monocular Perception of Biological Motion in Johansson Displays," *Computer Vision and Image Understanding,* vol. 81, pp. 303-327, 2001.

[28] Y. Song, "A Probabilistic Approach to Human Motion Detection and Labeling," PhD thesis, Caltech, 2003.

[29] N. Srebro, "Maximum Likelihood Bounded Tree-Width Markov Networks," *Proc. 16th Conf. Uncertainty in Artificial Intelligence,* pp. 504-511, 2001.

[30] C. Tomasi and T. Kanade, "Detection and Tracking of Point Features," *Technical Report CMU-CS-91-132, Carnegie Mellon Univ.,* 1991.

[31] S. Wachter and H.-H. Nagel, "Tracking Persons in Monocular Image Sequences," *Computer Vision and Image Understanding,* vol. 74, pp. 174-192, 1999.

[32] M. Weber, M. Welling, and P. Perona, "Unsupervised Learning of Models for Recognition," *Proc. European Conf. Computer Vision,* vol. 1, pp. 18-32, June/July 2000.

[33] M. Weber, "Unsupervised Learning of Models for Object Recognition," PhD thesis, Caltech, May 2000.

[34] M. Welling, "EM-Algorithm," Class Notes at California Inst. of Technology, 2000.

[35] Y. Yacoob and M.J. Black, "Parameterized Modeling and Recognition of Activities," *Computer Vision and Image Understanding,* vol. 73, pp. 232-247, 1999.

Yang Song received the BE (with honors) and ME degrees from the Department of Automation in Tsinghua University, China, in 1993 and 1996, respectively, and the MS and PhD degrees in electrical engineering from the California Institute of Technology, Pasadena, in 1998 and 2003, respectively. From 1992 to 1996, she worked for the National Key Laboratory of Pattern Recognition, Tsinghua University, China, as a research assistant. She is currently a research scientist at Fujifilm Software, Inc. in San Jose, California. Her research interests include machine learning and pattern recognition, computer vision, graphical models, signal and image processing, and data analysis. She is a member of the IEEE and the IEEE Computer Society.

Luis Goncalves received the BASc degree in electrical engineering from the University of Waterloo, Canada, in 1991. In 1992, he received the MS degree in electrical engineering, and in 2000, the PhD degree in computation and neural systems, both from the California Institute of Technology, Pasadena. For the past two years, he has held the position of research scientist at Idealab, Pasadena, California, where he does R&D work for start-up companies within Idealab. His most recent accomplishment has been as the principal investigator responsible for the robust, real-time, vision-based localization techniques utilized in Evolution Robotics' (an Idealab company) break-through visual-SLAM technology. Previously to joining Idealab, he co-founded Vederi Corp. and Realmoves Inc., companies which developed technology for efficiently scanning panoramic views of the streets of the entire urban US, and to synthesize realistic human character animation in real-time, respectively. He is the author of three US patents based on these three technologies. His research interests lie in the areas of machine learning, robotic vision, human-machine interfaces, human motion estimation and synthesis, and speech processing. He is a member of the IEEE and the IEEE Computer Society.

Pietro Perona received the degree of Dottore in ingegneria elettronica from the University of Padova, Italy in 1985, and the PhD degree in electrical engineering and computer science from the University of California, Berkeley, in 1990. He is currently a professor of electrical engineering and computation and neural systems, as well as the Director of the US National Science Foundation Engineering Research Center for Neuromorphic Systems Engineering (CNSE) at the California Institute of Technology, Pasadena. His research interests include both human and machine vision. His current activity is focused on visual recognition and the perception of 3D shape. He has also worked on the use of diffusive PDEs for image processing (anisotropic diffusion) and filtering techniques for early vision and modeling of human vision. He is a member of the IEEE and the IEEE Computer Society.