Learning Probabilistic Structure for Human Motion Detection

Yang Song†, Luis Goncalves†, and Pietro Perona†‡
† California Institute of Technology, 136-93, Pasadena, CA 91125, USA
‡ Università di Padova, Italy
{yangs,luis,perona}@vision.caltech.edu

Abstract

Decomposable triangulated graphs have been shown to be efficient and effective for modeling the probabilistic spatio-temporal structure of brief stretches of human motion. In previous work such model structure was hand-crafted by expert human observers and labeled data were needed for parameter learning. We present a method to build automatically the structure of the decomposable triangulated graph from unlabeled data. It is based on maximum-likelihood. Taking the labeling of the data as hidden variables, a variant of the EM algorithm can be applied. A greedy algorithm is developed to search for the optimal structure of the decomposable model based on the (conditional) differential entropy of variables. Our algorithm is demonstrated by learning models of human motion completely automatically from unlabeled real image sequences with clutter and occlusion. Experiments on both motion captured data and grayscale image sequences show that the resulting models perform better than the hand-constructed models.

1. Introduction

Humans are the most important component of a machine's environment. Detecting and interpreting human presence, actions and activities is one of the most valuable functions of our own visual system. Endowing machines with the same ability would enable a great number of useful industrial applications ranging from convenient non-contact user interfaces for consumer products, to on-board safety systems for automobiles, and surveillance systems for stores and museums.

A system for interpreting human activity must, first of all, be able to *detect* human presence [17, 16]. A second important task is to localize the visible parts of the body and assign appropriate labels to the corresponding regions of the image – for brevity we call this the *labeling* task [17, 16]. Given a labeling the different parts of the body may be *tracked* in time [15, 14, 2, 8, 9, 3, 19, 7]. Their trajectories and/or spatiotemporal energy pattern will allow a classification of the actions and activities [13, 21].

We focus here on detection and labeling. This problem was studied in the context of a 'generalized Johansson problem' [17]. The position and velocity of point-features is the input to a system that decides whether human motion is present. The system also assigns probabilistic labels (the main parts of the body plus a generic background label) to the detected features. The method is shown to be fast and robust both to extraneous clutter and to undetected body parts [17]. In [16], the algorithm is demonstrated to work well on a number of simple grayscale image sequences.

While the previous work is highly successful it is limited in scope: while the parameters of the probability density function at the heart of the model are estimated from training data, the 'triangulated' structure of such density function is hand-crafted. This is unsatisfactory for two reasons: first, it is time-consuming to develop such models by hand; second, the data should dictate such structure rather than the judgment of a human operator. Furthermore, the correct labelings for the training data are required to be known, which could be hard to obtain in practice. For example, the ground truth labeling for the training data in [16] is hand-constructed.

We address here the problem of unsupervised learning of model structure. We restrict our attention to triangulated models, since they both account for much correlation between the random variables that represent the position and motion of each body part, and they yield efficient algorithms. Our goal is to learn the best triangulated model, i.e., the one that reaches maximum likelihood with respect to the training data. We approach the problem in two settings: when the training features are *labeled*, i.e., the parts of the model and the correspondence between the parts and observed features are known (e.g. by a motion-capture system), and when the training features are *unlabeled*, i.e., the training features include both useful foreground parts and background clutter and the correspondence between the parts and detected features are unknown (e.g. when they

0-7695-1272-0/01 $10.00 © 2001 IEEE

are acquired with a monocular camera and no human intervention is practical). Our algorithm leads to systems able to learn models of human motion completely automatically from real image sequences - unlabeled training features with clutter and occlusion.

In section 2 we summarize the main facts about the triangulated probability model. In section 3 we address the labeled training set problem. In section 4 we address the unlabeled training set problem. In section 5 we present some experimental results, both on motion-captured data and on grayscale image sequences.

2. Decomposable triangulated graphs

Discovering the probability structure (conditional independence) among variables is important since it makes efficient learning and testing (labeling and detection for example) possible, hence some computationally intractable problems become tractable. Trees are good examples of modeling conditional (in)dependence [4, 12, 10]. A decomposable triangulated graph [1] is another type of graph which has been demonstrated to be useful for biological motion detection and labeling [17, 16].

A decomposable triangulated graph [1] is a collection of cliques of size three, where there is an elimination order of vertices such that when a vertex is deleted, it is only contained in one triangle and the remaining subgraph is again a collection of triangles until only one triangle left. Decomposable triangulated graphs are more powerful than trees since each node can be thought of as having two parents. Similarly to trees, efficient algorithms allow fast calculation of the maximum likelihood interpretation of a given set of data.

Conditional (in)dependences among random variables (parts) can be described by a decomposable triangulated graph. Let $\mathcal{S} = \{S1, S2, \ldots, SM\}$ be the set of M parts, and X_{Si}, $1 \leq i \leq M$, is the measurement for Si. If the joint probability density function $P(X_{S1}, X_{S2}, \ldots, X_{SM})$ can be decomposed as a decomposable triangulated graph, it can be written as,

$$\begin{aligned} & P_{whole}(X_{S1}, X_{S2}, \ldots X_{SM}) \\ = & \prod_{t=1}^{T} P_{A_t|B_tC_t}(X_{A_t}|X_{B_t}, X_{C_t}) \\ & \cdot P_{B_TC_T}(X_{B_T}, X_{C_T}) \end{aligned} \quad (1)$$

where $A_i, B_i, C_i \in \mathcal{S}$,
$1 \leq i \leq T = M - 2$,
$\{A_1, A_2, \ldots, A_T, B_T, C_T\} = \mathcal{S}$, 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 an elimination order for the decomposable graph.

3. Optimization of the decomposable triangulated graph

Suppose $\mathcal{X} = \{\overline{X}^1, \overline{X}^2, \ldots, \overline{X}^N\}$ are i.i.d samples from a probability density function, where $\overline{X}^n = (X^n_{S1}, \ldots, X^n_{SM})$, $1 \leq n \leq N$, are labeled data. We want to find the decomposable triangulated graph G, such that $P(G|\mathcal{X})$ is maximized. $P(G|\mathcal{X})$ is the probability of graph G being the 'correct' one given the observed data $\mathcal{X}$. Here we use G to denote both the decomposable graph and the conditional (in)dependence depicted by the graph. By Bayes' rule, $P(G|\mathcal{X}) = P(\mathcal{X}|G)P(G)/P(\mathcal{X})$, therefore if we can assume the priors $P(G)$ are equal for different decompositions, then our goal is to find the structure G which can maximize $P(\mathcal{X}|G)$. From the previous section, a decomposable triangulated graph G is represented by $(A_1, B_1, C_1), (A_2, B_2, C_2), \ldots, (A_T, B_T, C_T)$, then $P(\mathcal{X}|G)$ can be computed as follows,

$$\begin{aligned} & \log P(\mathcal{X}|G) \\ = & \log P(\overline{X}^1, \overline{X}^2, \ldots, \overline{X}^N|G) \\ = & \sum_{n=1}^{N} \log P(\overline{X}^n|G) \\ = & \sum_{n=1}^{N} \Big(\sum_{t=1}^{T} \log P(X^n_{A_t}|X^n_{B_t}, X^n_{C_t}) \\ & \qquad + \log P(X^n_{B_T}, X^n_{C_T}) \Big) \end{aligned} \quad (2)$$

$$\begin{aligned} \cong & -N \cdot \sum_{t=1}^{T} h(X_{A_t}|X_{B_t}, X_{C_t}) \\ & -N \cdot h(X_{B_T}, X_{C_T}) \end{aligned} \quad (3)$$

where $h(\cdot)$ is differential entropy or conditional differential entropy [5] of continuous random variables. Equation (3) is an approximation which converges to equality for $N \to \infty$ due to the weak Law of Large numbers and definitions and properties of differential entropy [5, 4, 6, 11, 12]. 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 the above equations can be maximized. If graphs with different elimination orders are taken as having different structures, then the total number of possible structure is $M!/2 \cdot \prod_{j=1}^{M-3}(2j+1)$, which makes exhaustive search only possible for small Ms. In our application $M > 10$ and therefore the number of graph structures is larger than 3×10^{12}.

Though for tree cases, the optimal structure can be obtained efficiently by the maximum spanning tree algorithm [4, 12], for decomposable triangulated graphs, there is no existing algorithm which runs in polynomial time and guarantees to the optimal solution. We develop a greedy algorithm to grow the graph by the property of decomposable graphs. For each possible choice of C_T (the last

vertex of the last triangle), find the best B_T which can maximize $-h(X_{B_T}, X_{C_T})$, then get the best child of edge (B_T, C_T) as A_T, i.e., the vertex (part) that can maximize $-h(X_{A_T}|X_{B_T}, X_{C_T})$. The next vertex is added one by one to the existing graph by choosing the best child of all the edges (legal parents) of the existing graph 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.

Let G_{exist} denote the decomposable graph obtained so far and V_{avail} denote the set of unused vertices (vertices to be added to the graph). The initial value for G_{exist} is a empty graph, and the initial value for V_{avail} is the set of composed parts $\mathcal{S}$. The algorithm can be described as following,

For each $C_T \in \mathcal{S}$,
  add C_T to G_{exist}
  remove C_T from V_{avail}
  for each $v \in V_{avail}$
    compute $-h(C_T, v)$
  find $B_T = \arg\max_{v \in V_{avail}} -h(C_T, v)$
  add vertex B_T and edge (B_T, C_T) to G_{exist}
  remove B_T from V_{avail}
  for each t from T to 1,
    for each edge $e \in G_{exist}$,
      for each $v \in V_{avail}$,
        compute $-h(v|e(1), e(2))$
      find $v^*(e) = \arg\max_v -h(v|e(1), e(2))$
    find $e_{sel} = \arg\max_e -h(v^*(e)|e(1), e(2))$
    let $A_t = v^*(e_{sel})$, $B_t = e_{sel}(1)$, and $C_t = e_{sel}(2)$
    add vertex A_t and edges (A_t, B_t), (A_t, C_t) to G_{exist}
    remove A_t from V_{avail}

From all the graphs originated from different C_T, choose the one with the highest $\log P(\mathcal{X}|G)$.

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, $M-2 = T \geq t \geq 1$, the number of edges in G_{exist} (legal parents) is $2*(T-t)+1$ and the number of vertices in V_{avail} (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. The algorithm is a greedy algorithm, with no guarantee that the global optimal solution could be found. Its effectiveness will be explored through experiments.

4. Unsupervised learning of the decomposable graph

In this section, we consider the case when only unlabeled data are available. Assume we have a data set of N samples $\mathcal{X} = \{\overline{X}^1, \overline{X}^2, \ldots, \overline{X}^N\}$. Each sample $\overline{X}^n$, $1 \leq n \leq N$, is a group of detected features which contains the target object, but $\overline{X}^n$ is unlabeled, which means the correspondence between the candidate features and the parts of the object is unknown. For example when we run a feature detector (such as Lucas-Tomasi-Kanade detector [18]) on real image sequences, the detected features can be from target objects and background clutter with no identity attached to each feature. We want to select the useful composite parts of the object and learn the probability structure from $\mathcal{X}$.

If the labeling for each $\overline{X}^n$ is taken as a hidden variable, then the EM algorithm can be used to learn the probability structure and parameters. We used a method similar to [20], but here all the candidate features are with the same type. 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 taken a value from $\mathcal{S} \cup \{BG\}$ (BG is the background clutter label). The observations for the EM algorithm 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 (i.e. the decomposable triangulated graph) and parameters for its associated probability density function. We use G to represent both the probability structure and the parameters. If we assume that $\overline{X}^n$s are independent from each other and h_n only depends on $\overline{X}^n$, then the likelihood function to maximize is,

$$\begin{aligned} L &= \log P(\mathcal{X}, G) \\ &= \log P(\mathcal{X}|G) + \log P(G) \\ &= \sum_{n=1}^N \log P(\overline{X}^n|G) + \log P(G) \\ &= \sum_{n=1}^N \log \sum_{h_{ni} \in H_n} P(\overline{X}^n, h_n = h_{ni}|G) + \log P(G) \end{aligned} \tag{4}$$

where h_{ni} is the ith possible labeling for $\overline{X}^n$, and H_n is the set of all such labelings. Optimization directly over equation (4) is hard, and the EM algorithm solves the optimization problem iteratively. In EM, for each iteration t, we will optimize the function,

$$\begin{aligned} & Q(G_t|G_{t-1}) \\ = \; & E[\log P(\mathcal{X}, \mathcal{H}, G_t)|\mathcal{X}, G_{t-1}] \\ = \; & \sum_{n=1}^N E[\log P(\overline{X}^n, h_n, G_t)|\overline{X}^n, G_{t-1}] \\ = \; & \sum_{n=1}^N \sum_{h_{ni} \in H_n} P(h_n = h_{ni}|\overline{X}^n, G_{t-1}) \\ & \qquad \cdot \log P(\overline{X}^n, h_n = h_{ni}, G_t) \\ = \; & \sum_{n=1}^N \sum_{h_{ni} \in H_n} R_{ni} \log P(\overline{X}^n, h_n = h_{ni}, G_t) \end{aligned} \tag{5}$$

where R_{ni} is the probability of $h_n = h_{ni}$ given the observation $\overline{X}^n$ and the decomposable probability structure G_{t-1}. For each iteration t, R_{ni} is a fixed number for a hypothesis h_{ni}. R_{ni} can be computed as,

$$\begin{aligned} & R_{ni} \\ = \; & P(h_{ni}|\overline{X}^n, G_{t-1}) \\ = \; & P(h_{ni}, \overline{X}^n, G_{t-1}) / \sum_{h_{ni}} P(h_{ni}, \overline{X}^n, G_{t-1}) \end{aligned} \quad (6)$$

We will discuss the computation of $P(h_{ni}, \overline{X}^n, G_{t-1})$ below. Under the labeling hypothesis $h_n = h_{ni}$, $\overline{X}^n$ is divided into the foreground features $\overline{X}^n_{fg}$, which are parts of the object, and background (clutter) $\overline{X}^n_{bg}$. If the foreground features $\overline{X}^n_{fg}$ are independent of clutter $\overline{X}^n_{bg}$, then,

$$\begin{aligned} & P(h_{ni}, \overline{X}^n, G) \\ = \; & P(\overline{X}^n|h_{ni}, G)P(h_{ni}, G) \\ = \; & P(\overline{X}^n_{fg}|h_{ni}, G)P(\overline{X}^n_{bg}|h_{ni}, G)P(h_{ni}|G)P(G) \end{aligned} \quad (7)$$

For simplicity, we will assume the priors $P(h_{ni}|G)$ are the same for different h_{ni}, and $P(G)$ are the same for different graph structures. If we assume uniform background densities like in [20, 17], then $P(\overline{X}^n_{bg}|h_{ni}, G)$ is the same for different h_{ni}. Under probability decomposition G, $P(\overline{X}^n_{fg}|h_{ni}, G)$ can be computed as in equation (1). Therefore the maximization of equation (5) is equivalent to maximizing,

$$\begin{aligned} & Q(G_t|G_{t-1}) \\ \sim \; & \sum_{n=1}^{N} \sum_{h_{ni}} R_{ni} \log[P(\overline{X}^n_{fg}|h_{ni}, G_t)] \\ = \; & \sum_{n=1}^{N} \sum_{h_{ni}} R_{ni} [\sum_{t=1}^{T} \log P(X^{ni}_{A_t}|X^{ni}_{B_t}, X^{ni}_{C_t}) \\ & + \log P(X^{ni}_{B_T}, X^{ni}_{C_T})] \end{aligned} \quad (8)$$

For most problems, the number of possible labelings is very large (on the order of n_k^M), so it is computationally prohibitive to sum over all the possible h_{ni} as in equation (8). However, if there is one hypothesis labeling h^*_{ni} that is much better than other hypotheses,, i.e. R^*_{ni} corresponding to h^*_{ni} is much larger than other R_{ni}'s, then R^*_{ni} can be taken as 1 and other R_{ni}'s as 0. Hence equation (8) can be approximated as,

$$\begin{aligned} Q(G_t|G_{t-1}) \sim \sum_{n=1}^{N} [\sum_{t=1}^{T} & \log P(X^{ni*}_{A_t}|X^{ni*}_{B_t}, X^{ni*}_{C_t}) \\ & + \log P(X^{ni*}_{B_T}, X^{ni*}_{C_T})] \end{aligned} \quad (9)$$

where $X^{ni*}_{A_t}, X^{ni*}_{B_t}$ and$X^{ni*}_{C_t}$ are measurements corresponding to the best labeling h^*_{ni}. Comparing equation (9) with equation (2), we know that for iteration t, if the best hypothesis h^*_{ni} 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^*_{ni} 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 vs. mixture of Gaussian for clustering problems.

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 step: for each $\overline{X}^n$, use G_{t-1} to find the best labeling h^*_{ni};
M step: use the data labeled with h^*_{ni} for each $\overline{X}^n$ to run the greedy graph growing algorithm described in section 3 and get G_t.

So far we assume that all the composed parts are observed. In the case of some parts missing (e.g. occlusion), the measurements for the missing parts can also be taken as hidden variables ([20]), and the above algorithm can be easily modified to handle the missing parts.

5. Experiments

We tested our algorithm on both motion capture data (Johansson displays) as in [17] and on detected features from real image sequences as in [16]. The motion capture data allowed us to run the learning algorithm under conditions where all body parts were present and their position in space was tracked with millimetric precision. The real image sequences presented a more challenging scenario where a two-frame noisy feature detector [18] was used to generate the training set, and with many occlusions occurring.

5.1. Results on motion capture data

We first investigate the performance of the algorithm on motion capture data as in [17]. The data consist of the 3-D positions of 14 markers fixed rigidly on a subject's body. These positions were tracked at 60Hz with 1mm accuracy as the subject walked back and forth for four minutes (two minutes are used for training, and the other two for testing). The 3-D data was projected to 2-D with a fixed orthographic projection so that the majority of the walking was seen from a 45 degree angle viewpoint (slightly from the side). Although the motion capture system provided labeled data, the data were treated as unlabeled for this experiment, and the labeling was only used as a ground truth to quantify the accuracy of the learned model.

We chose to learn models with 9 parts instead of all 14 to see if the model was able to consistently pick out 9 parts

and ignore the other 5. We assumed all the pdfs to be Gaussian, and the differential entropies can be computed from the covariance matrix [5]. We ran the EM-like algorithm described in section 4 ten times with different random initializations, and Figure 1(a) and (b) are the two best models obtained (with the highest likelihoods). The figure shows the mean positions of each model part (up to some horizontal and vertical scale factor), which corresponds quite nicely to the geometrical structure of the human body. The labels corresponding to each point were obtained by putting the original data's labels in correspondence with the results from the model. In the first model (a), the same vertex represents both the left and right knee (LK(RK)) (it detected the left knee 63% of the time and the right knee 37% of the time). This is due to the fact that, from an orthographic side view with all points present (i.e., no self-occlusions), during some parts of the walk cycle it is very difficult to distinguish the left and right knee, and so the model has accumulated the statistics of both into one point. A similar situation occurs with the ankles, point LA(RA). Since except for LK(RK) and LA(RA), each learned model part corresponds consistently to a 'real' body part (according to the ground truth labeling of the training set, see Figure 1), we can quantify the detection and labeling performance in testing.

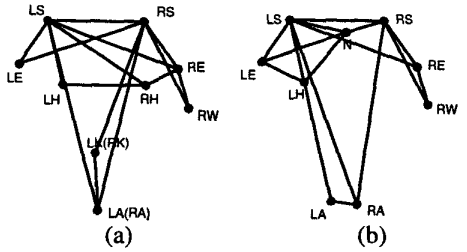


Figure 1. Two decomposable triangulated models for Johansson displays. These models were learned automatically from unlabeled training data. 'L': left; 'R': right. H:head, N:neck, S:shoulder, E:elbow, W:wrist, H:hip, K:knee, A:ankle.

Figure 2 shows the detection and labeling results by using the two models in Figure 1. Figure 2 (a) and (b) are ROC curves corresponding to Figure 1 (a) and (b) respectively. They were generated by comparing the likelihood of the model on frames consisting of only 30 random background points to frames with 30 background points plus 3 to 8 body parts present. With 5 or more body parts present, the ROC curve is nearly perfect. The dashed curve is the overall ROC considering all the frames used (from 3 to 8 body parts). The threshold corresponding to $P_{Detect} = 1 - P_{FalseAccept}$ on this curve was used for later experiments. The stars ('*') on the solid curves are corresponding to that threshold. Figure 2(c) shows the the detection rate vs. number of body parts displayed with regard to the fixed threshold. Figure 2 (d) is the curve of correct label rate (label-by-label rate) vs. number of body parts when a person is correctly detected. In Figure 2 (c) and (d), the solid lines (with *) are from model Figure 1 (a); the dashed lines (with o) are from model Figure 1 (b); and dash-dot lines with triangles are from the hand-crafted model in [17] (Figure 10 in that paper). We can conclude from this figure that the automatically learned models work quite well.

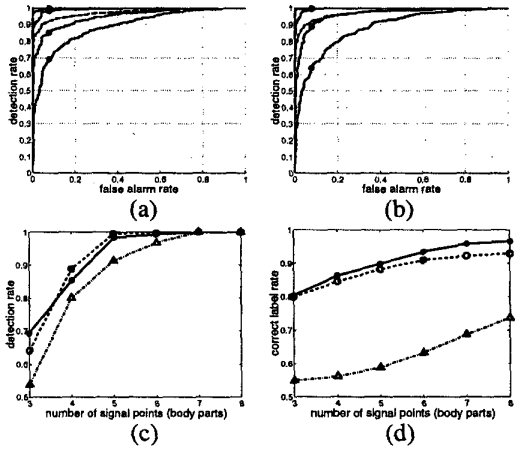


Figure 2. Detection and labeling results. (a) and (b) are ROC curves corresponding to models Figure 1 (a) and (b) respectively. Solid lines: 3 to 8 body parts with 30 background points vs. 30 background points only. The more body body parts present, the better the ROC. Dashed line: overall ROC considering all the frames used. The threshold corresponding to $P_D = 1 - P_{FA}$ on this curve was used for later experiments. The stars ('*') on the solid curves are corresponding to that threshold. (c) detection rate vs. number of body parts displayed with regard to the fixed threshold. (d) correct label rate (label-by-label rate) vs. number of body parts when a person is correctly detected. In (c) and (d), Solid lines (with *) are from model Figure 1 (a); dashed lines (with o) are from model Figure 1 (b); and dash-dot lines with triangles are from the hand-crafted model in [17] (Figure 10 in that paper).

5.2. Results on real image sequences

In this experiment we used the same image sequences as in [16]. Figure 3 shows sample frames of the data. There are three different types of motion: 1: A subject walks from the left back corner to the right front corner, facing about 60 degrees away from the front view (second row of Figure 3). For this motion, we have about 1000 frames (8 sequences, around 120 frames each) as the training set, and another 1500 frames (12 sequences) as the testing set. 2: A chair moves from left to right, about 1000 frames (8 sequences) in total. 3: While a subject walks, a chair also moves as a background moving object (first row of Figure 3). 2000 frames (16 sequences) were collected for motion type 3. The candidate features were obtained from a Lucas-Tomasi-Kanade algorithm [18] on two frames.

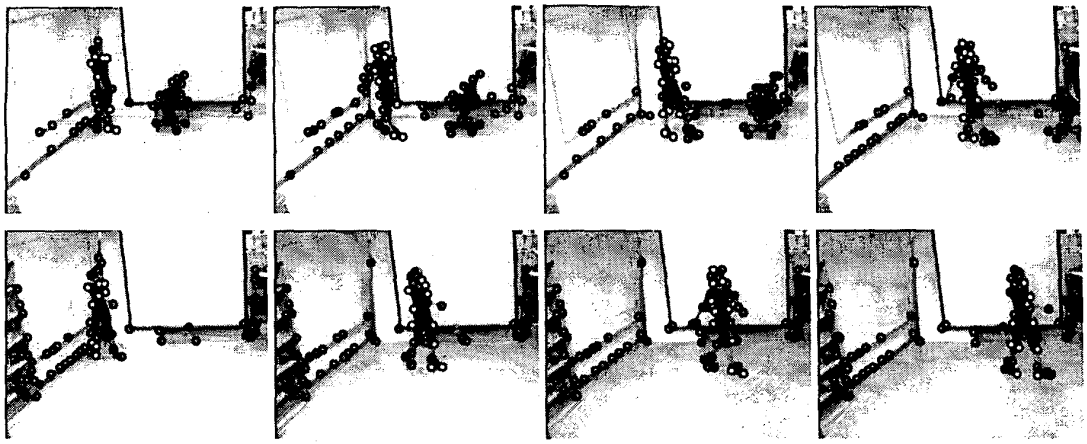


Figure 3. **Sample frames** from body and chair moving sequences (first row) and body moving sequences (second row). The dots (either in black or in white) are the features selected by Lucas-Tomasi-Kanade algorithm on two frames. The white dots are the most human-like configuration found by the automatically learned model (Figure 4).

For the training sequences we first did background subtraction, then ran L-T-K feature selector/tracker on two frames to get training features, because the background had persistent statistics (for example, some points along the white board were consistently detected). Background subtraction was necessary to avoid the unsupervised model from incorporating such background points with strong statistics. If the scenery of the dataset were more varied, so that the only detected points with persistent statistics were those on the human body, this background subtraction would not need to be done. The average number of detected training features per frame was 25. Furthermore, since in this real image sequence not all body features were present in each frame (due to self-occlusions or simply not being detected by the L-T-K detector), the algorithm described in section 4 was extended to handle the case of missing parts.

We learned an 11-feature model. Figure 4 shows the best model obtained after we ran the EM algorithms for 12 times. Figure 4(a) gives the mean positions and mean velocities (shown in arrows) of the composed parts selected by the algorithm. Figure 4(b) shows the learned decomposable triangulated probabilistic structure. The numbers in brackets show the correspondence of (a) and (b) and one elimination order.

The ROC curves in Figure 5 show the detection results. Detection is based on thresholding the likelihood of the most human-like configuration selected by the model. Solid lines are from the automatically learned model as in Figure 4; dashed lines are the from the model in [16] (dashed lines of Figure 7 in that paper). Figure 5(a) shows results of images with body and chair vs. images with chair only; and curves in Figure 5 (b) are results of images with body only vs. images with chair only. From Figure 5, we see that

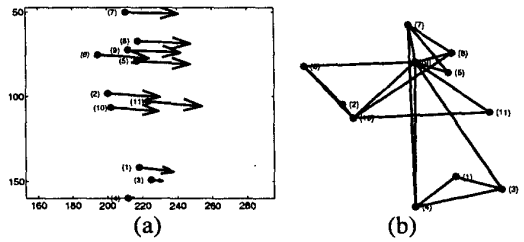
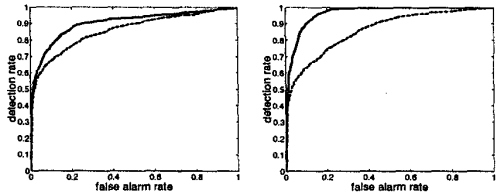


Figure 4. (a) The mean positions and mean velocities (shown in arrows) of the composed parts selected by the algorithm. (b) the learned decomposable triangulated probabilistic structure. The numbers in brackets show the correspondence of (a) and (b) and one elimination order.

the automatically learned model performs better than the hand-constructed model in [16]. The automatically learned model is also more efficient since there are only 11 parts in the model (there are 20 parts in the hand-constructed model in [16]).

6. Conclusions and Discussions

We have described a method for learning the structure and parameters of a decomposable triangulated graph in an unsupervised fashion from unlabeled data. We have applied this method to learn models of biological motion that can be used to detect and label reliably biological motion. When tested on real image sequences, the resulting model can detect the presence of biological motion with a significant improvement in accuracy over methods which rely on hand construction of model structure, and the model is also more efficient (with less number of composed parts). Our algo-



(a) (b)

Figure 5. **ROC curves.** (a) Results of images with body and chair vs. images with chair only. (b) Results of images with body only vs. images with chair only. Solid line: using the automatically learned model as in Figure 4; dashed line: using the model in [16] (dashed lines of Figure 7 in that paper).

rithm is EM-like. Like any EM method, our method is liable to be caught in local maxima. The use of a greedy algorithm for the maximization step (as well as the simplifying assumption that there is one labeling hypothesis that is much more likely than the rest) further increases the chance of running into a local maximum. Nevertheless, from the experiments we conducted it seems that a very good model can be found by running the algorithm on the same data ten times or so with different random initializations. Our algorithm enables the creation of systems that are able to learn models of human motion completely automatically from real image sequences. We intend to continue our work by systematically studying the trade-off between model complexity (number of vertices) and accuracy (we picked 9 for motion captured data and 11 for grayscale image sequences in our experiments in this paper), experimenting with different types of motions beyond the walking of a single subject, and developing methods of automatically learning a classifier of different types of motions from very long image sequences.

Acknowledgments

Funded by the NSF Engineering Research Center for Neuromorphic Systems Engineering (CNSE) at Caltech (NSF9402726), and by an NSF National Young Investigator Award to PP (NSF9457618).

References

[1] Y. Amit and A. Kong. Graphical templates for model registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:225–236, 1996.

[2] A. Blake and M. Isard. 3d position, attitude and shape input using video tracking of hands and lips. In *Proc. ACM Siggraph*, pages 185–192, 1994.

[3] C. Bregler and J. Malik. Tracking people with twists and exponential maps. In *Proc. IEEE CVPR*, pages 8–15, 1998.

[4] C. Chow and C. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14:462–467, 1968.

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

[6] N. Friedman and M. Goldszmidt. Learning bayesian networks from data. Technical report, AAAI 1998 Tutorial, http://robotics.stanford.edu/people/nir/tutorial/, 1998.

[7] D. Gavrila. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, 73:82–98, 1999.

[8] L. Goncalves, E. D. Bernardo, E. Ursella, and P. Perona. Monocular tracking of the human arm in 3d. In *Proc. 5^{th} Int. Conf. Computer Vision*, pages 764–770, Cambridge, Mass, June 1995.

[9] I. Haritaoglu, D. Harwood, and L. Davis. Who, when, where, what: A real time system for detecting and tracking people. In *Proceedings of the Third Face and Gesture Recognition Conference*, pages 222–227, 1998.

[10] S. Ioffe and D. Forsyth. Human tracking with mixtures of trees. In *International Conference on Computer Vision*, pages 690–695, July 2001.

[11] M. Jordan, editor. *Learning in Graphical Models*. MIT Press, 1999.

[12] M. Meila and M. Jordan. Learning with mixtures of trees. *Journal of Machine Learning Rearch*, 1:1–48, 2000.

[13] R. Polana and R. Nelson. Detecting activities. In *DARPA93*, pages 569–574, 1993.

[14] J. Rehg and T. Kanade. Digiteyes: Vision-based hand tracking for human-computer interaction. In *Proceedings of the workshop on Motion of Non-Rigid and Articulated Bodies*, pages 16–24, November 1994.

[15] K. Rohr. Incremental recognition of pedestrians from image sequences. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 8–13, New York City, June, 1993.

[16] Y. Song, X. Feng, and P. Perona. Towards detection of human motion. In *Proc. IEEE CVPR 2000*, volume 1, pages 810–817, June 2000.

[17] Y. Song, L. Goncalves, E. D. Bernardo, and P. Perona. Monocular perception of biological motion in johansson displays. *Computer Vision and Image Understanding*, 81:303–327, 2001.

[18] C. Tomasi and T. Kanade. Detection and tracking of point features. *Tech. Rep. CMU-CS-91-132,Carnegie Mellon University*, 1991.

[19] S. Wachter and H.-H. Nagel. Tracking persons in monocular image sequences. *Computer Vision and Image Understanding*, 74:174–192, 1999.

[20] M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for recognition. In *Proc. ECCV*, volume 1, pages 18–32, June/July 2000.

[21] Y. Yacoob and M. Black. Parameterized modeling and recognition of activites. *Computer Vision and Image Understanding*, 73:232–247, 1999.