A Probabilistic Approach to Human Motion Detection and Labeling

Thesis by

Yang Song

In Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

California Institute of Technology

Pasadena, California

2003

(Defended Nov 13, 2002)

Acknowledgements

First I would like to thank my advisor, Pietro Perona for admitting me into Caltech and for showing me what scientific research is all about. He played a very important rule in leading me towards scientific maturity. I am grateful to his support through the years on both scientific and personal matters.

I am grateful to my candidacy and defense committees, for serving on my committee, and for sharing their comments: Yaser Abu-Mostafa, Jehoshua Bruck, Richard Murray, Stefano Soatto, Jim Arvo, Mike Burl and Michelle Effros.

I am grateful to Luis Goncalves, my closest collaborator over several years. I benefited very much from many stimulating discussions with him and from his consistent encouragement. He is also very helpful in collecting the data set in chapter 6.

I am grateful to Xiaolin Feng and Enrico Di Bernardo for collaboration on the experiments in chapter 3 and for the motion capture data, to Charless Fowlkes for bringing structure learning problem to our attention and discussions on mixtures of trees, and to Max Welling for some inspiring discussions.

I would like to thank my fellow graduate students, Anelia Angelova, Christophe Basset, Arrigo Benedetti, Jean-Yves Bouguet, Domitilla Del Vecchio, Claudio Fanti, Rob Fergus, Pierre Moreels, Fei Fei Li, Mario Munich, Marzia Polito, and Silvio Savarese, for making the Vision Lab at Caltech a resourceful and pleasant place to work. I am grateful to the systems managers, Dimitris Sakellariou, Naveed Near-Ansari, Bob Freeman, Joseph Chiu, and Michael Potter, for making sure the computers working well. I am also grateful to Catherine Stebbins, Malene Hagen, Lavonne Martin, and Melissa Slemin for their help on administrative matters.

I would like to thank my friends outside the vision lab, Huayan Wang, Hong Xiao, Chengxiang (Rena) Yu, Qian Zhao, Yue Qi, Lifang Li, Hanying Feng, Tianxin Chen, Zhiwen Liu, Lu Sun, Xiaoyun Zhu, and Xubo Song for their help on various aspects during my graduate stay at Caltech.

Last, but certainly not the least, I would like to express my deepest gratitude to my family. I am grateful to my parents for their unconditional love and confidence in me, for their support during the hardest times, and for their patience during this long adventure. I am grateful to my husband, Xiao-chang, for his understanding and support, for his sacrifices to take extra family work, and for providing me with many everyday wisdoms. Finally, all of the work becomes meaningful because of my lovely daughter, Myra Miaobo, who has been very supportive by not crying much and giving me peace of mind. She motivates me to achieve more in life.

List of Publications

Work related to this thesis has been or will be presented in the following papers:

Unsupervised Learning of Human Motion,

Y. Song, L. Goncalves and P. Perona, submitted to IEEE Trans. on Pattern Analysis and Machine Intelligence.

Monocular Perception of Biological Motion in Johansson Displays,

Y. Song, L. Goncalves, E. Di Bernardo and P. Perona, Computer Vision and Image Understanding, vol. 81, no. 3, pages 303-327, 2001.

Learning Probabilistic Structure for Human Motion Detection,

Y. Song, L. Goncalves and P. Perona, Proc. IEEE Conf. Computer Vision and Pattern Recognition, vol. II, pages 771-777, December 2001.

Unsupervised Learning of Human Motion Models,

Y. Song, L. Goncalves and P. Perona, Advances in Neural Information Processing Systems 14, December 2001.

Towards Detection of Human Motion,

Y. Song, X. Feng and P. Perona, Proc. IEEE Conf. Computer Vision and Pattern Recognition, vol. I, pages 810-817, June, 2000.

Monocular perception of biological motion - clutter and partial occlusion,

Y. Song, L. Goncalves, and P. Perona, Proc. of 6th European Conferences on Computer Vision, vol. II, pages 719-733, June/July, 2000.

Monocular perception of biological motion - detection and labeling,

Y. Song, L. Goncalves, E. Di Bernardo and P. Perona, Proc. of 7th International

Conferences on Computer Vision, pages 805-812, September 1999.

A computational model for motion detection and direction discrimination in humans,

Y. Song and P. Perona, IEEE Computer Society Workshop on Human Motion, pages 11-16, December, 2000.

Abstract

Human motion analysis is a very important task for computer vision with many potential applications. There are several problems in human motion analysis: detection, tracking, and activity interpretation. Detection is the most fundamental problem of the three, but remains untackled due to its inherent difficulty. This thesis develops a solution to the problem. It is based on a learned probabilistic model of the joint positions and velocities of the body parts, where detection and labeling are performed by hypothesis testing on the maximum a posterior estimate of the pose and motion of the body. To achieve efficiency in learning and testing, a graphical model is used to approximate the conditional independence of human motion. This model is also shown to provide a natural way to deal with clutter and occlusion.

One key factor in the proposed method is the probabilistic model of human motion. In this thesis, an unsupervised learning algorithm that can obtain the probabilistic model automatically from unlabeled training data is presented. The training data include useful foreground features as well as features that arise from irrelevant background clutter. The correspondence between parts and detected features is also unknown in the training data. To learn the best model structure as well as model parameters, a variant of the EM algorithm is developed where the labeling of the data (part assignments) is treated as hidden variables. We explore two classes of graphical models: trees and decomposable triangulated graphs and find that the later are superior for our application. To better model human motion, we also consider the case when the model consists of mixtures of decomposable triangulated graphs.

The efficiency and effectiveness of the algorithm have been 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. We find detection rates of over 95% on pairs of frames. This is very promising for building a real-life system, for example, a pedestrian detector.

Contents

List of Figures

List of Tables

Chapter 1 Introduction

This thesis presents a new approach to human motion detection and labeling. In this chapter, we first give the motivation for this work, i.e., why the problem of human motion analysis is important and why this thesis focuses on detecting and labeling human motion. We then brief our approach and give the outline for the thesis.

1.1 Motivation for human motion analysis

Human motion analysis is an important but hard problem in computer vision. Humans are the most important component of our environment. Motion provides a large amount of information about humans and is very useful for human social interactions. The goal of human motion analysis is to extract information about human motion from video sequences. As shown in Figure 1.1, for a given video sequence, we want to develop a computer system/algorithm which can give us a description of the scene. The description should first address whether there are humans in the scene. If so, how many there are, where they are located, and what they are doing.

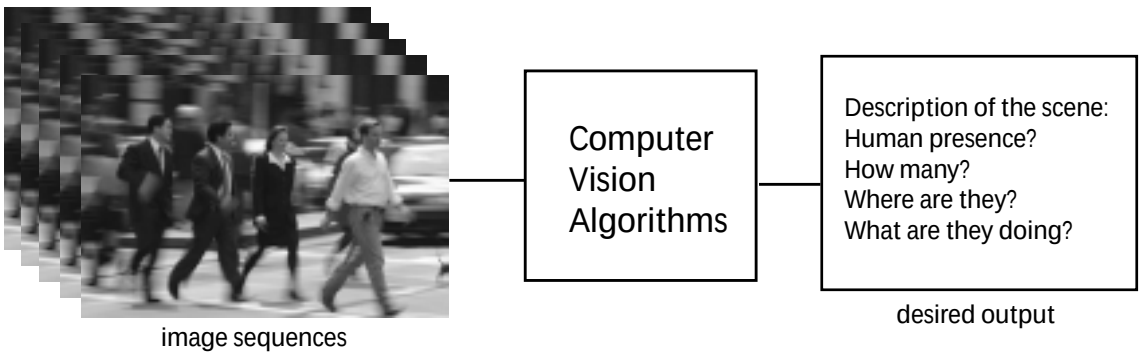


Figure 1.1: Human motion analysis.

Solving this problem can lead to many potential applications including but not

limited to:

- For the security of airports or big museums, it is very useful that a computer can detect automatically if someone is doing something suspicious, e.g., trying to grab a piece of art work.
- Human motion detection is also attractive to the automobile industry. Pedestrian detection is very important for transportation safety and for automated navigation.
- Human computer interfaces. We use keyboard, mouse and/or joystick as our input devices. If the computer could recognize what we mean when we point at it and/or give our instruction by our body movement, it would make the computer more user-friendly.

However, human motion analysis is difficult. First of all, the human body is richly articulated-even a simple stick model describing the pose of arms, legs, torso and head requires more than 20 degrees of freedom. The body moves in 3-D which makes the estimation of these degrees of freedom a challenge in a monocular setting [3, 4]. Image processing is also a challenge: humans typically wear clothing which may be loose and textured. This makes it difficult to identify limb boundaries, and even more so to segment the main parts of the body.

1.2 Problems in human motion analysis

A system for interpreting human activity must, first of all, be able to *detect* human presence. 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. Detection and labeling are coupled problems. Once we know the body parts assignments, we know the presence of a person; and vice versa. Given a labeling, different parts of the body may be *tracked* in time [5, 6, 7, 3, 8, 9, 10, 11]. Their trajectories and/or spatiotemporal energy pattern will allow a classification of the actions and activities [12, 13], which leads to *activity interpretation*.

Among these problems, activity interpretation needs to take the results of detection and tracking as input, whereas tracking algorithms need initializations, which can be provided by either detection, or in the absence of which, by ad hoc heuristics. Hence detection is the most fundamental problem of the three. In the field of computer vision, tracking has recently been an area of much attention, where considerable progress has been made. Detection, on the contrary, remains an open problem and will be the focus of this thesis.

1.3 Human perception: Johansson experiments

Our work on human motion detection and labeling is inspired by human perception. A striking demonstration of the capabilities of the human visual system is provided by the experiments of Johansson [14]. Johansson filmed people acting in total darkness with small light bulbs fixed to the main joints of their body. A single frame (Figure 1.2) of a Johansson movie is nothing but a cloud of identical bright dots on a dark field; however, as soon as the movie is animated, one can readily detect, count, segment a number of people in a scene, and even assess their activity, age, and sex [15, 16, 17]. Although such perception is completely effortless, our visual system is ostensibly solving a hard combinatorial problem (the labeling problem-which dot should be assigned to which body part of which person?).

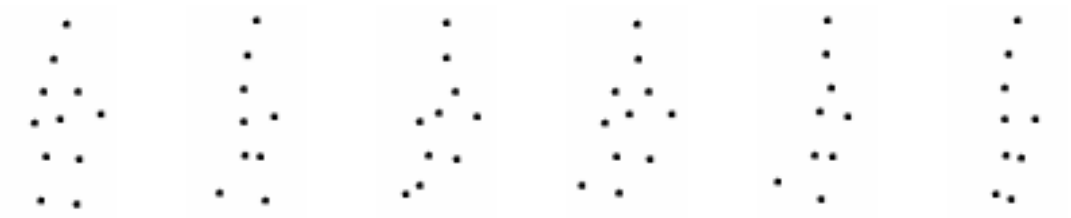


Figure 1.2: Sample frames of Johansson's display. In Johansson's original experiments, black background was used instead of white background.

Johansson experiments prove that motion is an important cue for visual perception. The fact that vivid motion can be perceived easily from a Johansson display

illustrates that our visual system has developed a very strong ability in perceiving human motion-we can recognize human motion easily from dots representing the motion of the main joints. This psychophysical evidence inspires us to build a computer algorithm to achieve what human eyes can do.

1.4 Approach

We believe that the human visual system gains the ability of recognizing body motion through learning (daily observation)*. Hence rather than modeling the details of the mechanics of the human body, we choose to approach human motion perception as the problem of recognizing a peculiar spatio-temporal pattern which may be learned perceptually. We approach the problem using learning and statistical inference.

We model how a person moves in a probabilistic way. Though different persons move in different styles and the same person moves differently at different times, a certain type of motion must share some common features. Moreover, the proportions of the body are in a similar range despite the difference in human body size. Hence a probabilistic model which captures both the common features and the variance of human motion is very appropriate.

The approach on gray-scale images is shown in Figure 1.3. To detect and label a moving human body, a feature detector/tracker (such as a corner detector) is first used to obtain candidate features from a pair of frames. The combination of features is then selected based on maximum likelihood by using the joint probability density function formed by the position and motion of the body. Detection is performed by thresholding the likelihood (see the lower part of Figure 1.3).

We use point features (from a motion capture system or a corner detector) because they are easier to obtain compared to other types of features, such as body segments, which may be more susceptible to occlusion. Point features are also a natural choice since psychophysics experiments (Johansson's experiments [14]) indicate that the hu-

*We once showed a movie of the top-view of one person walking, and it became much harder to recognize that it was a person walking. One reasonable explanation is that it is because we usually see a person walking from front view, side view, or back view, but not from the top.

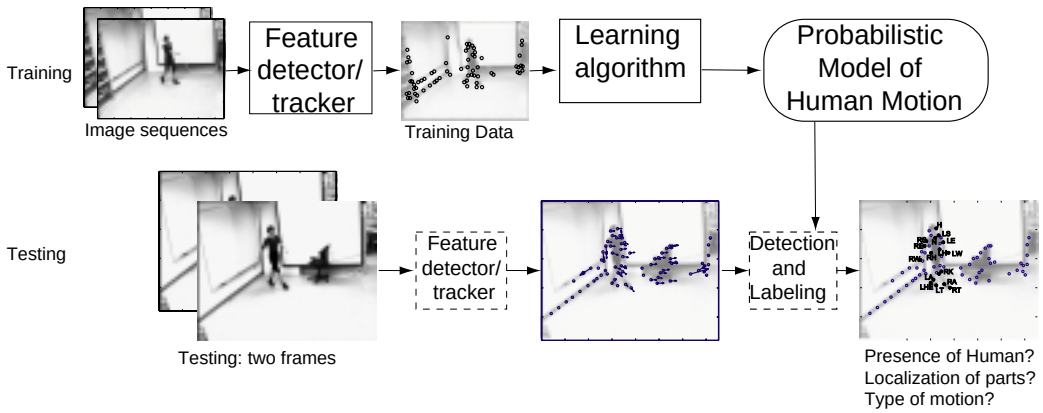


Figure 1.3: Diagram of the system on gray-scale images.

man visual system can perceive vivid human motion from moving dots representing the motion of the human body joints. However, this does not preclude the use of this algorithm to other types of features.

One key factor in the method is the probabilistic model of human motion. In order to avoid an exponential combinatorial search, a graphical model is used to depict the conditional independence of body parts. Graphical models are a marriage between probability theory and graph theory [18]. We originally apply them to the problem of human motion detection and labeling. We explore two classes of graphical models: trees and decomposable triangulated graphs and find that the latter are superior for our application.

At the training stage of our approach, probabilistic independence structures as well as model parameters are learned from a training set. There are two types of training data-*labeled* and *unlabeled*. In the case of labeled training data, the parts of the model and the correspondence between the parts and observed features in the training set are known, e.g., data from a motion capture system. For labeled training data, we can hand-craft the probabilistic independence structure and estimate the model parameters (e.g., mean and covariance for unimodal Gaussian). We use this learning method in Chapters 2 and 3. In Chapter 4, we tackle a more challenging learning problem, where algorithms are developed to search for the optimal independence structure from labeled training data.

In the case of unlabeled training data, probabilistic models are learned from training features including both useful foreground parts and background clutter, and the correspondence between the parts and detected features is unknown. The problem arises when we run a feature detector (such as the Lucas-Tomasi-Kanade detector [1]) on real-image sequences, features are detected both on target objects and background clutter with no identity attached to each feature. From these features, we wish to know which feature combinations arise in correspondence to a given visual phenomenon (e.g., person walking from left to right). In Chapters 5 and 6, we develop unsupervised algorithms that are able to learn models of human motion completely automatically from real image sequences, i.e., unlabeled training features with clutter and occlusion.

1.5 Outline of the thesis

This thesis is organized as follows.

Chapter 2 considers the problem of labeling a set of observed points when there is no clutter and no body parts are missing, which we call the 'Johansson problem.'

Chapter 3 explains how to extend the algorithm to perform detection and labeling in a cluttered and occluded scene, which we call the 'generalized Johansson problem.'

Chapter 4 describes how to learn the conditional independence structure of the probabilistic model from *labeled* data.

Chapter 5 addresses the learning problem when the training features are *unlabeled.*

Chapter 6 introduces the concept of mixtures of decomposable triangulated models and extends the unsupervised learning algorithm to the mixture model. This chapter also presents a more comprehensive experimental section than previous chapters.

Chapter 7 puts decomposable triangulated models in the general framework of graphical models, compares them with trees, and justifies the use of decomposable triangulated graphs.

Chapter 8 summarizes the thesis work and indicates possible future research directions.

Chapter 2 The Johansson problem

In Johansson's human perception experiments, the input to the human visual system are moving dots, and we can get a vivid perception of human motion and assign body parts (such as hand, elbow, shoulder, knee and foot) to the dots immediately [14]. During this process, our visual system has solved a hard combinatorial problem-the *labeling* problem: which dot should be assigned to which body part of which person? This chapter develops an algorithm providing a solution to the labeling problem when there is no clutter and no body parts are missing. Since the display is very similar to that of Johansson's experiments, we call it the 'Johansson problem.'

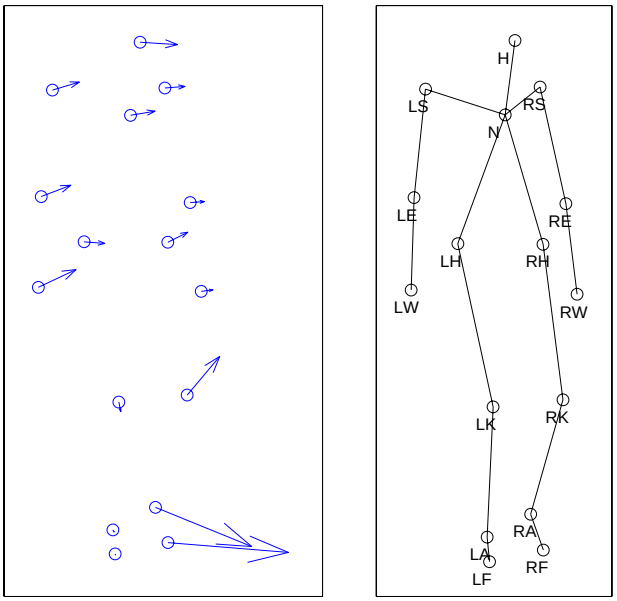
2.1 Notation and approach

As shown in Figure 2.1, given the position and velocity (arrows in the figure) of some dots* in the image plane (Figure 2.1 (a)), we want to assign the correct labels to the dots. Velocity is used to characterize the motion. In our Johansson scenario each part appears as a single dot in the image plane. Therefore, its identity is not revealed by cues other than its relative position and velocity.

We deploy a probabilistic approach. The body pose and motion are characterized by the joint probability density of the position and velocity of its parts. Let $\mathcal{S}_{body} = \{LW, LE, LS, H \ldots RF\}$ be the set of M body parts, for example, LW is the left wrist, RF is the right foot, etc. Correspondingly, let X_{LW} be the vector representing the position and velocity of the left wrist, X_{RF} be the vector of the right foot, etc. We model the pose and motion of the body probabilistically by means of a probability density function $P_{\mathcal{S}_{body}}(X_{LW}, X_{LE}, X_{LS}, X_H, \ldots, X_{RF})$.

Suppose that there are N point features in a display. Let $\overline{X} = [X_1, \ldots, X_N]$ be

*In this thesis, the words, 'dots,' 'points,' 'markers,' 'features' or 'point features,' have the same meaning: things observed from the images. We will use them interchangeably. The words, 'parts' or 'body parts', mean the parts that compose of the object (a moving human in our application).



(a) (b)

Figure 2.1: The labeling problem (without clutter and missing points): given the position and velocity of body parts in the image plane (a), we use a probabilistic model to assign the correct labels to the body parts (b). '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.

the vector of measurements (each X_i, $i = 1, \ldots, N$, is a vector describing position and velocity of point i). Here we 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 labeling problem is to find $\overline{L}^*$, over all possible label vectors $\overline{L}$, such that the posterior probability of the labeling given the observed data is maximized, that is,

$$\overline{L}^* = \arg\max_{\overline{L} \in \mathcal{L}} P(\overline{L}|\overline{X}) \tag{2.1}$$

where $P(\overline{L}|\overline{X})$ is the conditional probability of a labeling $\overline{L}$ given the data $\overline{X}$ and $\mathcal{L}$ is the set of all possible labelings. Using Bayes' law:

$$P(\overline{L}|\overline{X}) = P(\overline{X}|\overline{L}) \frac{P(\overline{L})}{P(\overline{X})} \tag{2.2}$$

It is reasonable to assume that the priors $P(\overline{L})$ are equal for different labelings, then

$$\overline{L}^* = \arg\max_{\overline{L}\in\mathcal{L}} P(\overline{X}|\overline{L}) \tag{2.3}$$

Given a labeling $\overline{L}$, each point feature i has a corresponding label L_i. Therefore each measurement X_i may also be written as X_{L_i}, i.e., the measurement corresponding to a specific body part associated with label L_i. For example, if $L_i = LW$, i.e., the label corresponding to the left wrist is assigned to the ith point, then $X_i = X_{LW}$ is the position and velocity of the left wrist. Then,

$$P(\overline{X}|\overline{L}) = P_{\mathcal{S}_{body}}(X_{LW}, X_{LE}, X_{LS}, X_H, \ldots, X_{RF}) \tag{2.4}$$

where $P_{\mathcal{S}_{body}}$ is the joint probability density function of the position and velocity of all the M body parts.

Three problems face us at this point: (a) What is the structure for the probability/likelihood function to be maximized? (b) How do we estimate its parameters? (c) How do we reduce the computational cost of the combinatorial search problem of finding the optimal labeling? Problems (a) and (c) need to be addressed together: the structure of the probability density function must be such that it allows efficient optimization.

A brute force solution to the optimization problem is to search exhaustively among all $M!$ (assuming no clutter, no missing body parts) possible $\overline{L}$'s and find the best one. The search cost is factorial with respect to M. Assume $M = 16$, then the number of possible labelings is larger than 2×10^{13}, which is computationally prohibitive.

It is useful to notice that the body is a kinematic chain: for example, the wrist is connected to the body indirectly via the elbow and the shoulder. One could assume that the position and the velocity of the wrist are, therefore, independent of the

position and velocity of the rest of the body once the position and velocity of elbow and shoulder are known. This intuition may be generalized to the whole body: once the position and velocity of a set S of body parts is known, the behavior of body parts that are separated by S is independent. Of course, this intuition is only an approximation which needs to be validated experimentally.

Our intuition on how to decompose the problem may be expressed in the language of probability: consider the joint probability density function of 5 random variables $P(A, B, C, D, E)$. By Bayes' rule, it may be expressed as $P(A, B, C, D, E) = P(A, B, C)P(D|A, B, C)P(E|A, B, C, D)$. If these random variables are conditionally independent as described in the graph of Figure 2.5, then

$$P(A, B, C, D, E) = P(A, B, C)P(D|B, C)P(E|C, D) \tag{2.5}$$

Thus, if the body parts can satisfy the appropriate conditional independence conditions, we can express the joint probability density of the pose and velocity of all parts as the product of conditional probability densities of n-tuples. This approximation makes the optimization step computationally efficient as will be discussed below.

What is the best decomposition for the human body? What is a reasonable size n of the groups (or cliques) of body parts? We hope to make n as small as possible to minimize the cost of the optimization. But as n gets smaller, conditional independence may not be a reasonable approximation any longer. There is a tradeoff between computational cost and algorithm performance. We use decomposable triangulated models with $n = 3$ as will be discussed below.

2.2 Decomposable triangulated graphs

We use *decomposable triangulated* graphs[†] to depict the probabilistic conditional independence structure of body parts. A decomposable triangulated graph [19] is a

[†]For general graphical models, the term *decomposable* and the term *triangulated* have their own meanings (they are actually equivalent properties[18]). In this thesis, we use the term *decomposable triangulated* specifically for the graph type defined in this paragraph.

collection of cliques[‡] 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); (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 left.

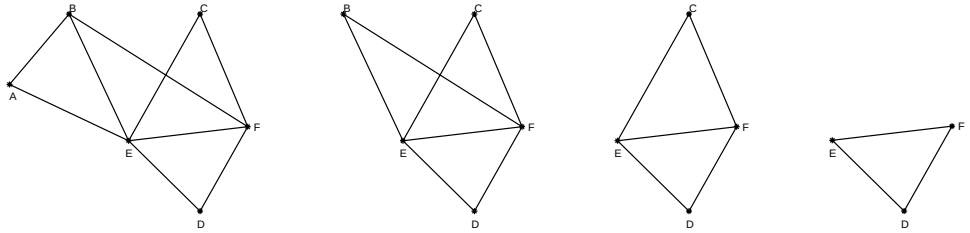


Figure 2.2: Example of successive elimination of a decomposable triangulated graph, with elimination order $(A, B, C, (DEF))$.

Figure 2.2 shows an example of a decomposable triangulated graph. The cliques of the graphs are $\{A, B, E\}$, $\{B, E, F\}$, $\{C, E, F\}$, and $\{D, E, F\}$. One elimination order of the vertices is A, B, C, and $\{D, E, F\}$ is left as the last clique. Figure 2.2 gives the steps of elimination of vertices following this order. Note that for a fixed graph structure, the elimination order is not unique. For example, for the graph in Figure 2.2, another elimination order of vertices is C, D, F with $\{A, B, E\}$ left as the last clique.

Figure 2.3 shows two decomposable graphs of the whole body, along with one order of successive elimination of the cliques.

To better understand the concept of the decomposable triangulated graph, some graphs which are not decomposable triangulated graphs are given in Figure 2.4. They are not decomposable triangulated graphs for the followings reasons. Figure 2.4 (a): after one free vertex and its associated edges are deleted, the remaining graph is not a collection of cliques of size three; Figure 2.4 (b): there is no free vertex in the graph; Figure 2.4 (c): it is a clique of size four, not a collection of cliques of size three.

When decomposable graphs are used to describe conditional independence of ran-

[‡]A clique is a maximal subset of vertices, any two of which are adjacent.

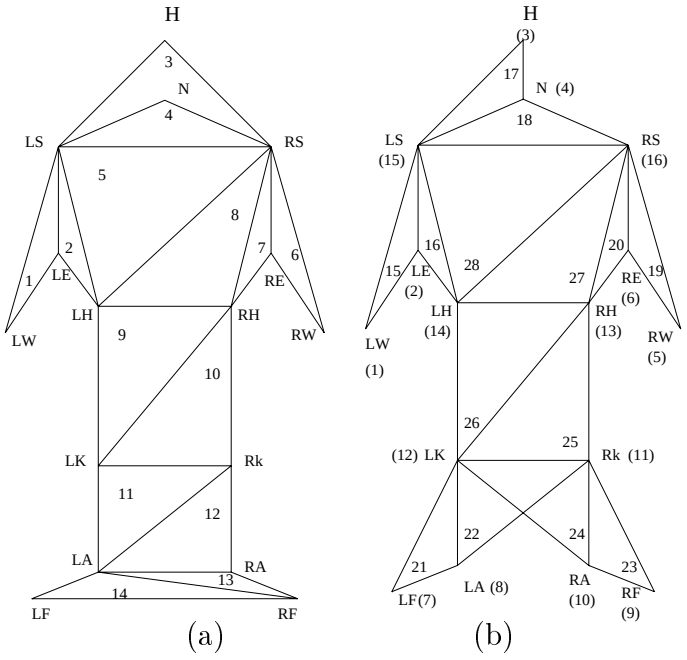


Figure 2.3: Two decompositions of the human body into triangles. 'L' and 'R' in label names indicate left and right. H:head, N:neck, S:shoulder, E:elbow, W:wrist, H:hip, K:knee, A:ankle and F:foot. The numbers inside triangles give the index of triangles used in the experiments. In (a) they are also one order in which the vertices are deleted. In (b) the numbers in brackets show one elimination order.

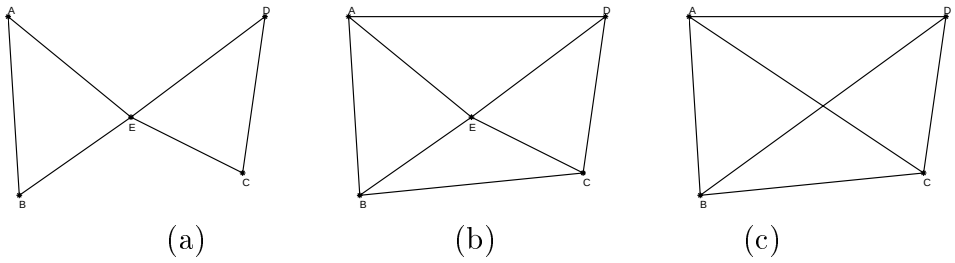


Figure 2.4: Examples of non-decomposable triangulated graphs.

dom variables, the probability density function can be written according to the elimination order of the vertices. For example, following the elimination order given in Figure 2.2, the joint probability $P(A, B, C, D, E, F)$ can be approximated by

$$P(A, B, C, D, E, F) = P(A|B, E)P(B|E, F)P(C|E, F)P(D, E, F) \tag{2.6}$$

If we use another elimination order mentioned above, C, D, F with $\{A, B, E\}$ left as the last clique, then the joint probability $P(A, B, C, D, E, F)$ can be written as

$$P(A, B, C, D, E, F) = P(C|E, F)P(D|E, F)P(F|B, E)P(A, B, E) \tag{2.7}$$

Using Bayes' rule, it is easy to verify that equations (2.6) and (2.7) are equivalent. For one graph, although we can write different decompositions according to different elimination orders, they describe the same conditional independence.

In general, Let $\mathcal{S}_{body} = \{S1, S2, \ldots, SM\}$ be the set of M parts, for example, $S1$ denotes the left wrist, SM is the right foot, etc. X_{Si}, $1 \leq i \leq M$, is the measurement for Si. If the joint probability density function $P_{\mathcal{S}_{body}}$ can be decomposed as a decomposable triangulated graph, it can be written as

$$\begin{aligned} & P_{\mathcal{S}_{body}}(X_{S1}, X_{S2}, \ldots X_{SM}) \\ = & \prod_{t=1}^{T-1} P_{A_t|B_tC_t}(X_{A_t}|X_{B_t}, X_{C_t}) \cdot P_{A_TB_TC_T}(X_{A_T}, X_{B_T}, X_{C_T}) \end{aligned} \tag{2.8}$$

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

The choice of decomposable triangulated graph is motivated by both computational and performance reasons. Trees are good examples of modeling conditional (in)dependence [20, 21]. But decomposable triangulated graphs are more powerful models than trees since each node can be thought of as having two parents. Similar to trees, decomposable triangulated graphs allow efficient algorithms such as dynamic programming to fast calculate the maximum likelihood interpretation of a given set

of data [19]. We will give more rigorous analysis on why we choose decomposable triangulated graphs in section 7.5. The details of the dynamic programming algorithm will be discussed in the next section.

2.3 Algorithms

What is needed is an algorithm that will search through all the legal labelings and find the one that maximizes the global joint probability density function. Notice that this optimum cannot be obtained by optimizing independently each triplet (clique of size three). If the joint probability can by decomposed by a decomposable triangulated graph, dynamic programming can be used to solve this problem efficiently. The key condition for using dynamic programming is that the problem exhibits optimal substructure. For example, we want to find the labeling which can maximize $P(A,B,C,D,E)$. If equation (2.5) holds, then whatever the choices of A,B,C,D are, the best E must be the one which maximizes $P(E|C,D)$. Therefore to get the best E, we only need to consider the function $P(E|C,D)$ instead of $P(A,B,C,D,E)$. More formally,

$$\begin{aligned}\max_{A,B,C,D,E} P(A,B,C,D,E) &= \max_{A,B,C}(P(A,B,C)\cdot\max_{D}(P(D|B,C)\cdot\max_{E}P(E|C,D))) \\ &= \max_{A,B,C}(P(A,B,C)\cdot\max_{D}(f(B,C,D))) \\ &= \max_{A,B,C} g(A,B,C) \qquad (2.9)\end{aligned}$$

where $f(B,C,D) = P(D|B,C)\cdot\max_E P(E|C,D)$ and $g(A,B,C) = P(A,B,C)\cdot\max_D f(B,C,D)$. Assume each variable can take N possible values. If the maximization is performed over $P(A,B,C,D,E)$ directly, then the size of the search space is N^M (M is the number of variables, $M=5$ for this example). By equation (2.9), the maximization can be achieved by maximization over $P(E|C,D)$, $f(B,C,D)$ and $g(A,B,C)$ successively, and the size of the search space is $(M-2)\cdot N^3$.

Generally, if the joint probability of the whole body can be decomposed as in

equation (2.8), then

$$\begin{aligned} & \max P_{\mathcal{S}_{body}}(X_{S1}, X_{S2}, \ldots X_{SM}) \\ = & \max_{X_{A_T}, X_{B_T}, X_{C_T}} P_T(X_{A_T}, X_{B_T}, X_{C_T}) \max_{X_{A_{T-1}}} P_{T-1}(X_{A_{T-1}} | X_{B_{T-1}}, X_{C_{T-1}}) \cdots \\ & \max_{X_{A_2}} P_2(X_{A_2} | X_{B_2}, X_{C_2}) \max_{X_{A_1}} P_1(X_{A_1} | X_{B_1}, X_{C_1}) \end{aligned} \tag{2.10}$$

where the 'max' operation is computed from right to left.

If we take the probability density function as the cost function, a dynamic programming method similar to that described in [19] can be used. For each triplet (A_t, B_t, C_t), we characterize it with a ten dimensional feature vector

$$\mathbf{x} = (v_{Ax}, v_{Bx}, v_{Cx}, v_{Ay}, v_{By}, v_{Cy}, p_{Ax}, p_{Cx}, p_{Ay}, p_{Cy})^T \tag{2.11}$$

The first three dimensions of $\mathbf{x}$ are the x-direction (horizontal) velocity of body parts (A_t, B_t, C_t), the next three are the velocity in the y-direction (vertical), and the last four dimensions are the positions of body parts A_t and C_t relative to B_t. Relative positions are used here so that we can obtain translation invariance. As a first-order approximation, it is convenient to assume that $\mathbf{x}$ is jointly Gaussian-distributed and therefore its parameters may be estimated from training data using standard techniques. After the joint probability density function is computed, the conditional one can be obtained accordingly:

$$P_{A_t|B_tC_t}(X_{A_t} | X_{B_t}, X_{C_t}) = \frac{P_{A_tB_tC_t}(X_{A_t}, X_{B_t}, X_{C_t})}{P_{B_tC_t}(X_{B_t}, X_{C_t})} \tag{2.12}$$

where $P_{B_tC_t}(X_{B_t}, X_{C_t})$ can be obtained by estimating the joint probability density function of the vector $(v_{Bx}, v_{Cx}, v_{By}, v_{Cy}, p_{Cx}, p_{Cy})^T$.

Let

$$\Psi_t(X_{A_t}, X_{B_t}, X_{C_t}) = \log P_{A_t|B_tC_t}(X_{A_t} | X_{B_t}, X_{C_t}), \ for\ 1 \leq t \leq T-1 \tag{2.13}$$

$$\Psi_t(X_{A_t}, X_{B_t}, X_{C_t}) = \log P_{A_T B_T C_T}(X_{A_T}, X_{B_T}, X_{C_T}), \qquad for\ t = T \qquad (2.14)$$

be the cost function associate with each triangle, then the dynamic programming algorithm can be described as follows:

Stage 1: for every pair (X_{B_1}, X_{C_1}),

Compute $\Psi_1(X_{A_1}, X_{B_1}, X_{C_1})$ for all possible X_{A_1}

Define $T_1(X_{A_1}, X_{B_1}, X_{C_1})$ the total value so far.

Let $T_1(X_{A_1}, X_{B_1}, X_{C_1}) = \Psi_1(X_{A_1}, X_{B_1}, X_{C_1})$

Store $\begin{cases} X^*_{A_1[X_{B_1}, X_{C_1}]} = \arg\max_{X_{A_1}} T_1(X_{A_1}, X_{B_1}, X_{C_1}) \\ T_1(X^*_{A_1[X_{B_1}, X_{C_1}]}, X_{B_1}, X_{C_1}) \end{cases}$

Stage t, $2 \leq t \leq T$: for every pair (X_{B_t}, X_{C_t}),

Compute $\Psi_t(X_{A_t}, X_{B_t}, X_{C_t})$ for all possible X_{A_t}

Compute the total value so far (till stage t):

– Define $T_t(X_{A_t}, X_{B_t}, X_{C_t})$ the total value so far.

Initialize $T_t(X_{A_t}, X_{B_t}, X_{C_t}) = \Psi_t(X_{A_t}, X_{B_t}, X_{C_t})$

– If edge (A_t, B_t) is contained in a previous

stage and τ is the latest such stage, add the cost

$T_\tau(X^*_{A_\tau[X_{A_t}, X_{B_t}]}, X_{A_t}, X_{B_t})$ (or $T_\tau(X^*_{A_\tau[X_{B_t}, X_{A_t}]}, X_{B_t}, X_{A_t})$ if the

edge was reversed) to $T_t(X_{A_t}, X_{B_t}, X_{C_t})$

– Likewise, add the costs of the latest previous

stages containing respectively edge (A_t, C_t) and edge (B_t, C_t)

to $T_t(X_{A_t}, X_{B_t}, X_{C_t})$

Store $\begin{cases} X^*_{A_t[X_{B_t}, X_{C_t}]} = \arg\max_{X_{A_t}} T_t(X_{A_t}, X_{B_t}, X_{C_t}) \\ T_t(X^*_{A_t[X_{B_t}, X_{C_t}]}, X_{B_t}, X_{C_t}) \end{cases}$

When stage T calculation is complete, $T_T(X^*_{A_T[B_T,C_T]}, X_{B_T}, X_{C_T})$ includes the value of each Ψ_t, $1 \leq t \leq T$, exactly once. Since the Ψ_t's are the logs of conditional (and joint) probabilities, then if equation (2.8) holds,

$$T_T(X^*_{A_T[B_T,C_T]}, X_{B_T}, X_{C_T}) = \log P_{\mathcal{S}_{body}}(X_{LW}, X_{LE}, X_{LS}, X_H \ldots X_{RF})$$

Thus picking the pair $(X^*_{B_T}, X^*_{C_T})$ that maximizes T_T automatically maximizes the joint probability density function.

The best labeling can now be found tracing back through each stage: the best $(X^*_{B_T}, X^*_{C_T})$ determines $X^*_{A_T}$, then the latest previous stages with edge respectively $(X^*_{A_T}, X^*_{B_T})$, $(X^*_{A_T}, X^*_{C_T})$, and/or $(X^*_{B_T}, X^*_{C_T})$ determine more labels and so forth.

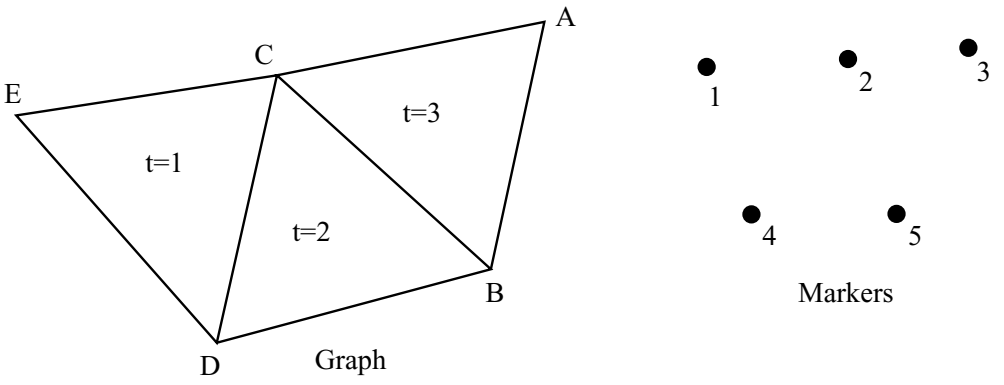
A simple example of this algorithm is shown in Figure 2.5.

The above algorithm is computationally efficient. Assume M is the number of body part labels and N ($N = M$ for this section) is the number of candidate markers, then the total number of stages is $T = M - 2$ and in each stage the computation cost is $\mathcal{O}(N^3)$. Thus, the complexity of the whole algorithm is on the order of $M * N^3$.

2.4 Experiments

We did experiments on motion capture data[§], which allow us to explore the labeling performance of the algorithm on frames with all the body parts observed and no clutter points. The data were obtained filming a subject moving freely in 3-D; 16 light bulbs were strapped to the main joints of the subject's body. In order to obtain ground-truth, the data were first acquired, reconstructed and labeled in 3-D using a 4-camera motion capture system operating at a rate of 60 samples/sec. Since our goal is to detect and label the body directly in the camera image plane, a generic camera view was simulated by orthographic projection of the 3-D marker coordinates. In the following sections we will control the camera view with the azimuth viewing angle: a value of 0 degrees will correspond to a right-side view, a value of 90 to a frontal

[§]These data were captured by Drs. Luis Goncalves and Enrico Di Bernado using a motion capture system built in Vision Lab, Caltech.



Stage 1: for all valid choices of markers for C and D (C=1..5, D=1..5, $C \neq D$)
for all valid choices of markers for E, (E=1..5, $E \neq C$, $E \neq D$)

compute $\Psi_1(E, C, D)$

Store $\begin{cases} E^*_{[C,D]} & E^*_{[C,D]} \text{ is best choice of E} \\ \Psi_1(E^*_{[C,D]}, C, D) & \text{for each choice of } C \text{ and } D \end{cases}$

Stage 2: for all valid choices of B and C
for all valid choices of D

compute $\Psi_2(D, B, C)$

let $T_2(D, B, C) = \Psi_2(D, B, C) + \Psi_1(E^*_{[C,D]}, C, D)$

Store $\begin{cases} D^*_{[B,C]} \\ T_2(D^*_{[B,C]}, B, C) \end{cases}$

Stage 3: for all valid choices of A and B
for all valid choices of C

compute $\Psi_3(C, A, B)$

let $T_3(C, A, B) = \Psi_3(C, A, B) + T_2(D^*_{[B,C]}, B, C)$

$$= \Psi_3(C, A, B) + \Psi_2(D^*_{[B,C]}, B, C) + \Psi_1(E^*_{[C,D^*_{[B,C]}]}, C, D^*_{[B,C]})$$

$$= \log(p(A, B, C)) + \log(p(D^*_{[B,C]}|B, C)) + \log(p(E^*_{[C,D^*_{[B,C]}]}|C, D^*_{[B,C]}))$$

$$= \log(p(A, B, C, D^*_{[B,C]}, E^*_{[C,D^*_{[B,C]}]}))$$

Thus chosing A,B, and C that maximizes T_3 finds the solution which maximizes the joint probability of the entire graph.

Figure 2.5: An example of dynamic programming algorithm applied to a simple graph. The goal is to assign the markers to the variables A, B, C, D, E in the graph such that $P(A, B, C, D, E)$ is maximized.

view of the subject. Six sequences were acquired each around 2 minutes long. In the next sections they will be referred as follows: Sequences W1 (7000 frames), W2 (7000 frames): relaxed walking forward and backwards along almost straight paths (with ± 20 degree deviations in heading); W3 and W4 (6000 frames each): relaxed walking, with the subject turning around now and then (Figure 2.6(a) shows sample frames from W3); Sequence HW (5210 frames): walking in a happy mood, moving the head, arms, hips more actively (Figure 2.6(b)); Sequence DA (3497 frames): dancing and jumping (Figure 2.6(c)), with the subject moving his legs and arms freely and much faster than in the previous four sequences. Given that the data were acquired from the same subject and that orthographic projection was used to simulate a camera view, our data were already normalized in scale. The velocity of each candidate marker was obtained by subtracting its positions in two consecutive frames. Thus, to get velocity information, we assumed that features could be tracked for two frames but we didn't use any feature correspondence over more than two frames, which is arguably the most difficult conditions under which to perform labeling and detection, as will be discussed in section 3.3.

Among the sequences, walking sequences W1 and W2 are the relatively simple ones, so W1 and W2 were first used to test the validity of the Gaussian probabilistic model and the performance of two possible body decompositions (Figure 2.3). Since the heading direction of W1 and W2 was roughly along a line, these sequences were also used to study the performance as a function of viewing angle. Then experiments were conducted using W3, HW and DA to see how the model worked for more active and non-periodic motions.

2.4.1 Detection of individual triangles

In this section, the performance of the Gaussian probabilistic model for individual triangles is examined. In the training phase, the joint Gaussian parameters (mean and covariance) for each triangle in Figure 2.3 were estimated from walking sequence W1 (viewed with a 45 degrees viewing angle). In the test phase, for each frame

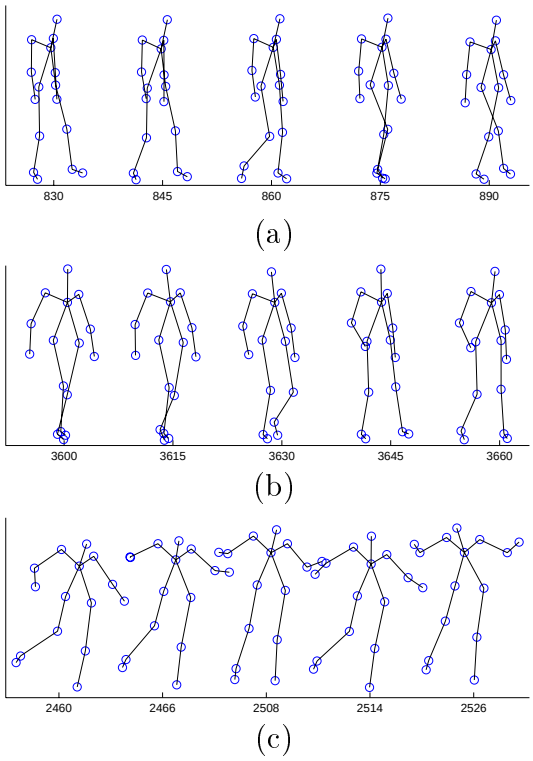
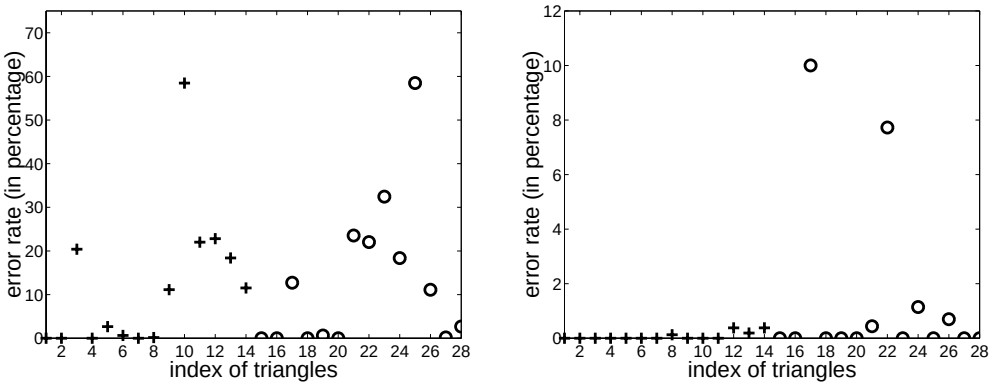


Figure 2.6: Sample frames for the (a) walking sequence W3; (b) happy walking sequence HW; (c) dancing sequence DA. The numbers on the horizontal axes are the frame numbers.

in W2 (also viewed of 45 degrees), each triangle probability was evaluated for all possible combinations of markers ($16 \times 15 \times 14$ different combinations). Ideally, the correct combination of markers should produce the highest probability for each respective triangle. Otherwise, an error occurred. Figure 2.7 (a) shows how well each triangle's joint probability model detects the correct set of markers. Figure 2.7 (b) shows a similar result for the conditional probability densities of triangles, where for each triangle conditional probability density $P_{A_t|B_tC_t}(X_{A_t}|X_{B_t}, X_{C_t})$, we computed $P_{A_t|B_tC_t}(X_{A_t}|X_{B_t}, X_{C_t})$ for all the possible choices of A_t (14 choices), given the correct choice of markers for B_t and C_t. Figure 2.7 shows that the Gaussian model is very good for most triangles (in the joint case, if a triangle is chosen randomly, then the chance of getting the correct one is 3×10^{-4} and the probability models do much better than that).



(a) (b)

Figure 2.7: Local model error rates (percentage of frames for which the correct choice of markers did not maximize each individual triangle probability). Triangle indices are those of the two graph models of Figure 2.3. '+': results for decomposition Figure 2.3(a); 'o': results for decomposition Figure 2.3 (b). (a) joint probability model; (b) conditional probability model.

It is not surprising that the performance of some triplets is much worse than others. The worst triangles in Figure 2.7 (a) are those with left and right knees, which makes sense because the two knees are so close in some frames that it is even hard for human eyes to distinguish between them. Therefore, it is also hard for the probability model to make the correct choice.

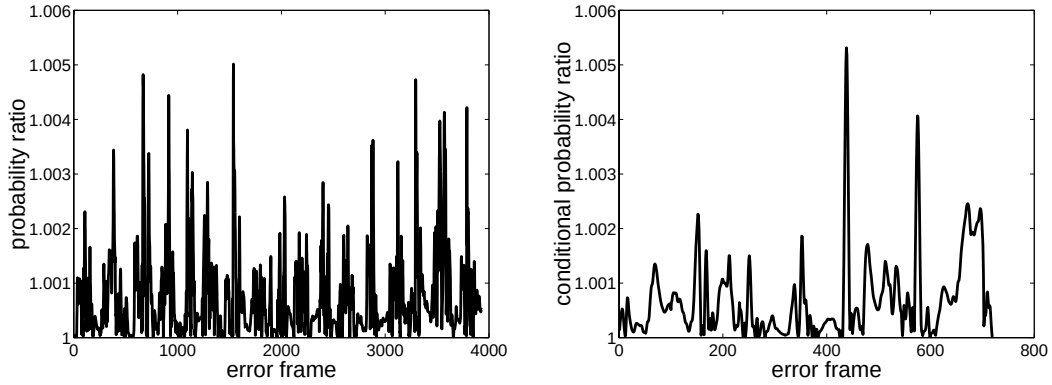


Figure 2.8: Probability ratio (correct markers vs. the solution with the highest probability when an error happens). The horizontal axis is the index of frames where error happens. (a) joint probability ratio for triangle 10 or 25 (RH, LK, RK); (b) conditional probability ratio for triangle 17 (H, N, LS).

Further investigation of the behavior of the triangle probabilities revealed that, for frames in which the correct choice of markers did not maximize a triangle probability, that probability was nevertheless quite close to the maximal value. Figure 2.8 shows the ratio of the probabilities of the correct choice over the maximizing choice for the two worst behaving triangles, over the set of frames where the errors occurred. Figure 2.8 (a) shows the ratio of the joint probability distribution for triangle 10 (consisting of right hip, left knee, and right knee, as in figure 2.3 (a)). Figure 2.8 (b) shows the ratio of the conditional probability distribution for triangle 17 (head, neck, and left shoulder). Although these two triangles had the highest error rates, the correct marker combination was always very close to being the highest ranking, always less than a factor of 1.006 away. This is a good indication that the individual triangle probability models encode the distribution quite well.

2.4.2 Performance of different body graphs

We did experiments using the two decompositions in Figure 2.3. The training sequence W1 and the test sequence W2 were under the same viewing angle: 45 degrees, which is between the side view and the front view. Table 1 shows the results. The

frame-by-frame error is the percentage of frames in which errors occurred, and *label-by-label error* is the percentage of markers wrongly labeled out of all the markers in all the testing frames. Label-by-label error is smaller than frame-by-frame error because an error in a frame does not mean all the markers are wrongly labeled.

decomposition model	(a)	(b)
frame-by-frame error	0.27%	13.13%
label-by-label error	0.06%	1.61%

Table 2.1: Error rates using the models in Figure 2.3

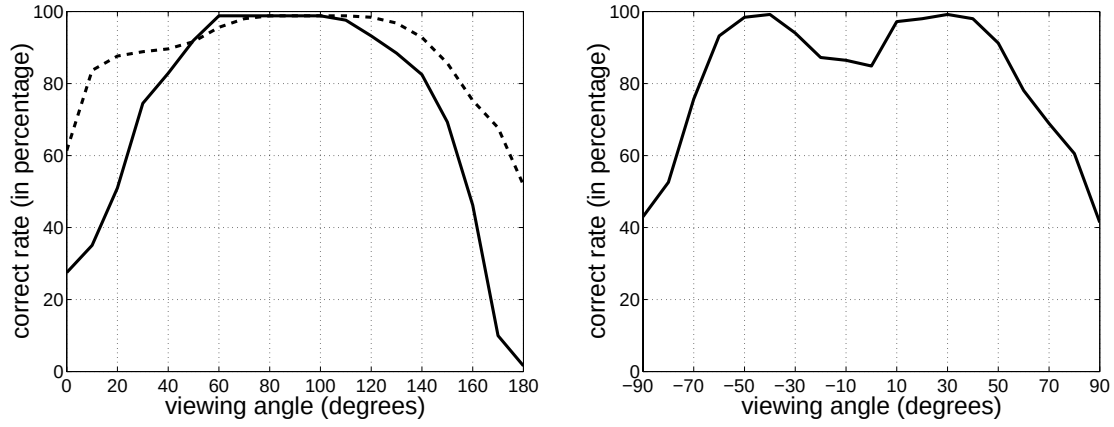
The performance of the algorithm using the decomposition of Figure 2.3(a) is almost perfect and much better than that of (b), which is consistent with our expectation (by Figure 2.7, the local performance of decomposition Figure 2.3(a) is better than that of Figure 2.3(b)). We used the better model in the rest of the experiments.

2.4.3 Viewpoint invariance

In the previous sections the viewing angle for training and for testing was the same. Here we explore the behavior of the method when the testing viewing angle is different from that used during training. Figure 2.9 shows the results of three such experiments where walking sequence W1 was used as the training set and W2 as the test set .

The solid line in Figure 2.9(a) shows the percentage of frames labeled correctly when the training was done at a viewing angle of 90 degrees (subject facing the camera) and the testing viewing angle was varied from 0 degrees (right-side view) to 180 degrees (left side view) in increments of 10 degrees. When the viewing angle was between 60 to 120 degrees, almost all frames were labeled correctly, thus showing that the probabilistic model learned at 90 degrees is insensitive to changes in viewpoint by up to ± 30 degrees.

The solid line in Figure 2.9(b) shows the results of a similar experiment where the training viewpoint was at 0 degrees (right-side view) and the testing angle was varied from -90 degrees (back view) to 90 degrees (front view) in 10 degree increments. A



(a) (b)

Figure 2.9: Labeling performance as a function of viewing angle. **(a)** Solid line: percentage of correctly labeled frames as a function of viewing angle, when the training was done at 90 degrees (frontal view). Dashed line: training was done by combining data from views at 30, 90, and 150 degrees. **(b)** Labeling performance when the training was done at 0 degrees (right-side view of walker). The dip in performance near 0 degrees is due to the fact that from a side view orthographic projection without body self-occlusions it is almost impossible to distinguish left and right.

noticeable dip in the performance centered around 0 degrees is visible in the plot. Inspection of the errors which occurred at these viewing angles revealed that they consisted solely of confusions between homologous left-right leg parts; i.e., the two hips were sometimes confused, as were the knees, the ankles, and the feet. Considering that an orthographic projection of the 3-D data was used to create the 2-D views, this result is not surprising; given an orthographic side view of a person walking (with no self-occlusions) a person viewing the motion is unable to distinguish the left and right-sides of the body. Thus, modulo this left-right ambiguity, the model learned at 0 degrees viewing angle is insensitive to changes in viewpoint of up to ± 50 degrees.

The dashed line in Figure 2.9(a) shows the results of an experiment of trying to increase the invariance of the probabilistic model with respect to changes in viewpoint. The same 3-D training sequence was used to generate three 2-D data sequences with viewing angles at 30, 90, and 150 degrees. The three 2-D sequences were combined, and used all together to learn the probability density functions of the graph triangles. As shown in the plot, this procedure does in fact improve the labeling accuracy. At

0 degrees, the only errors were the above mentioned left-right ambiguity within the legs. Between 10 and 60 degrees, besides left-right errors, also the feet and ankles were confused. From 120 to 180 degrees, the errors once again consisted solely of swapped left and right body parts.

2.4.4 Performance with different motions

The previous sections show that for simple motions very good results can be achieved using the probabilistic model. Here we want to investigate how the method works for more general sets of motions. We did experiments on walking sequence W3, happy walking sequence HW, and dancing sequence DA. Each sequence was divided into four segments for a total of twelve segments. To test a segment, frames from all the other eleven segments were used as the training set. The error rates for different sequences are obtained by averaging the results of the corresponding segments.

test set	ALL	W3	HW	DA
frame-by-frame error	6.81%	3.02%	4.49%	15.95%
label-by-label error	0.69%	0.38%	0.50%	1.45%

Table 2.2: Error rates for different sequences. ALL: average over all three sequences; W3: walking sequence; HW: walking in happy mood; DA: dancing sequence

Table 2 shows the error rates for different sequences. The first column is the average result for all three sequences, and the next three columns show the error rates for walking sequence W3, happy walking sequence HW and dancing sequence DA respectively. The results for walking sequence W3 and happy walking sequence HW are very good, with *frame-by-frame error* less than 5% and *label-by-label error* no more than 0.5%. It is not surprising that the error rates of dancing sequence are higher than the walking sequences because the motions in the dancing sequence are more random and agitated and therefore harder to model. Another possible reason is that the dancing sequence is shorter than the other sequences, so the motion of dancing has relatively less weight in the training set.

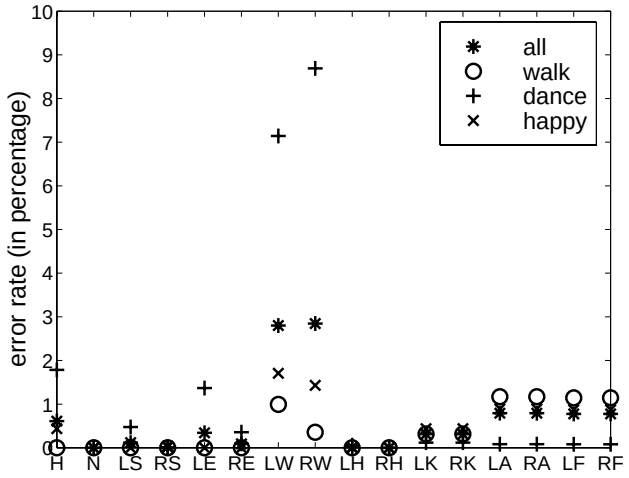


Figure 2.10: Error rates for individual body parts. 'L' and 'R' in label names indicate left and right. H:head, N:neck, S:shoulder, E:elbow, W:wrist, H:hip, K:knee,A:ankle and F:foot. See section 2.4.4.

Figure 2.10 shows the error rate of each individual body part for each of the sequences. Notice that most errors occur at the left and right wrist (LW and RW) in the dancing sequence. This is because in the dancing sequence the wrists are very close to hips in some frames, and the program mistook the hip markers as being the wrists. The reason why the program wouldn't mistake wrist markers as hips is that hips have better motion constraints than wrists. In our decomposed body graph Figure 2.3(a), both left and right hip (LH and RH) appear in five triangles, but the wrists (LW and RW) are only in one triangle each.

2.5 Summary

In this chapter, we develop an algorithm to solve the labeling problem with all the body parts present and no clutter, i.e., the 'Johansson problem.' We model the pose and motion of the body probabilistically by the joint probability density function (pdf) of the positions and velocities of all the body parts. Decomposable triangulated graphs are used to model the conditional independence of body parts so that dynamic programming can be used to find the best labeling efficiently. Experiments on motion capture data show that the algorithm works well for the 'Johansson problem.'

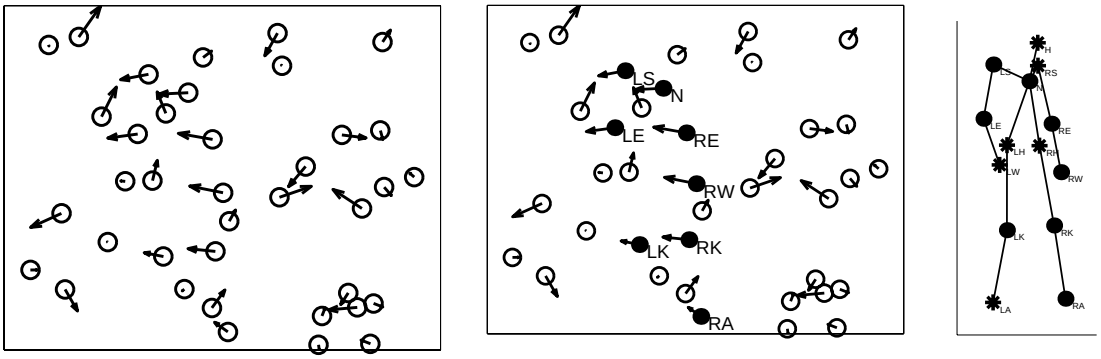
Chapter 3 Generalized Johansson problem: clutter and occlusion

In the previous chapter we dealt with the ideal case where all the body parts are present with no clutter points. But in real scenes, there is often clutter due to other moving patterns (cars driving by, trees swinging in the wind, water rippling... as in Figure 3.1) or the noisy output of feature detector/selector. Also, some body parts are not visible due to self-occlusion (Figure 3.1). In this chapter, we extend the algorithm to handle occlusion and clutter. We call the labeling and detection problem under clutter and occlusion 'generalized Johansson problem'.

Figure 3.1: Perception of biological motion in real scenes: one has to contend with a large amount of clutter (more than one person in the scene, other objects in the scene are also moving), and a large amount of self-occlusion (typically only half of the body is seen). Observe that segmentation (arm vs. body, left and right leg) is at best problematic.

The generalized Johansson problem can be formulated as follows: given the positions and velocities of many points in an image plane (Figure 3.2 (a)), we want to decide whether a human body is present (*detection*) and find the most likely human configuration (*labeling*) (Figure 3.2 (b)). In practice, the set of dots and associated

velocities can be obtained from a low-level motion detector/feature tracker applied to the entire image (for example, Lucas-Tomasi-Kanade feature detector/tracker [1]).



(a) (b) (c)

Figure 3.2: Detection and labeling under the conditions of clutter and occlusion: Given the position and velocity of dots in an image plane (a), we want to decide whether a person is present in the scene and find the most possible human configuration. Filled dots in (b) are body parts and circles are background points. Arrows in (a) and (b) show the velocities. (c) is the full configuration of the body. Filled (blackened) dots representing those present in (b), and the '*'s are actually missing (not available to the program). The body part label names are the same as in Figure 2.1.

In the following sections, we first address the labeling problem, i.e., how to find the most human-like configuration from a given set of features. Based on the tools and concepts developed for the labeling problem, we will describe how to do detection and count the number of people in the scene.

3.1 Labeling problem under clutter and occlusion

3.1.1 Notation and description of the problem

Similar to section 2.1, the labeling problem can be described as follows. Suppose that we observe N points (as in Figure 3.2(a), where $N = 38$). We assign an arbitrary

index to each point. Then,

$$
\begin{aligned}
i &\in 1,\ldots,N && \text{Index} && (3.1)\\
\overline{X} &= [X_1,\ldots,X_N] && \text{Vector of measurements} && (3.2)\\
\overline{L} &= [L_1,\ldots,L_N] && \text{Vector of labels} && (3.3)\\
L_i &\in \mathcal{S}_{body}\cup\{BG\} && \text{Possible values for each label} && (3.4)
\end{aligned}
$$

Since there exist clutter points that do not belong to the body, the background label BG is added to the label set. Due to clutter and occlusion, N is not necessarily equal to M (which is the size of $\mathcal{S}_{body}$). If we assume that the priors $P(\overline{L})$ are equal, then as in equation (2.3), we want to find

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

Let $\overline{\mathcal{L}}_{body}$ denote the set of body parts appearing in $\overline{L}$, $\overline{X}_{body}$ be the vector of measurements labeled as body parts, and $\overline{X}_{bg}$ be the vector of measurements labeled as background (BG). More formally, we group the measurements $\overline{X}$ in two vectors $\overline{X}_{body}$ and $\overline{X}_{bg}$,

$$
\begin{aligned}
\overline{\mathcal{L}}_{body} &= \{L_i; i=1,\ldots,N\}\cap\mathcal{S}_{body}\\
\overline{X}_{body} &= [X_{i_1},\ldots,X_{i_K}] \qquad \text{such that } \{L_{i_1},\ldots,L_{i_K}\}=\overline{\mathcal{L}}_{body}\\
\overline{X}_{bg} &= [X_{j_1},\ldots,X_{j_{N-K}}] \qquad \text{such that } L_{j_1}=\cdots=L_{j_{N-K}}=BG \qquad (3.5)
\end{aligned}
$$

where K is the number of points described in $\overline{X}_{body}$ (i.e. the size of $\overline{\mathcal{L}}_{body}$) and $N-K$ is the number of points in $\overline{X}_{bg}$, i.e. the number of background points.

If we assume that the position and velocity of the visible body parts is independent of position and velocity of clutter points, then,

$$P(\overline{X}|\overline{L}) = P_{\overline{\mathcal{L}}_{body}}(\overline{X}_{body})\cdot P_{bg}(\overline{X}_{bg}) \qquad (3.6)$$

where $P_{\overline{\mathcal{L}}_{body}}(\overline{X}_{body})$ is the marginalized probability density function of $P_{\mathcal{S}_{body}}$ (as in equation (2.4)) according to $\overline{\mathcal{L}}_{body}$. If independent uniform background noise is assumed, $P_{bg}(\overline{X}_{bg}) = (1/S)^{N-K}$, where $N-K$ is the number of background points, and S is the volume of the space the position and velocity of a background point lies in. In the following sections, we will address the issues of estimating $P_{\overline{\mathcal{L}}_{body}}(\overline{X}_{body})$ and further find the $\overline{L}^*$ with the highest likelihood.

3.1.2 Approximation of foreground probability density function

If no body part is missing, we can use equation (2.8) to get an approximation of the foreground probability density $P_{\overline{\mathcal{L}}_{body}}(\overline{X}_{body})$,

$$P_{\overline{\mathcal{L}}_{body}}(\overline{X}_{body}) = \prod_{t=1}^{T-1} P_t(X_{A_t}|X_{B_t}, X_{C_t}) \cdot P_T(X_{A_T}, X_{B_T}, X_{C_T}) \tag{3.7}$$

where T is the number of triangles in the decomposable triangulated graph, t is the triangle index, A_t is the first body part associated to triangle t, and etc.

If some body parts are missing, the foreground probability density function (PDF) is the marginalized version of the above equation – marginalization over the missing body parts. Let us consider the example in equation (2.5) and Figure 2.5. If A is missing, the marginalized PDF is $P(B, C, D, E)$, and,

$$P(B, C, D, E) = P(B, C) \cdot P(D|B, C) \cdot P(E|C, D) \tag{3.8}$$

But if C is missing, there is no conditional independence among variables A, B, D and E, and the marginalized PDF $P(A, B, D, E)$ cannot be decomposed into terms of smaller cliques. Hence the search cost for optimization is increased by one oder of magnitude. This exposes a general problem for precise marginalization. It may destroy some conditional independence and increase the computational cost.

We want the marginalization to be a good approximation of the true marginal PDF and allow efficient computation as well. A reasonable way to get such an ap-

proximation is to remove all the edges connected to the missing body parts, which may enforce stronger conditional independence. In formulas, this is equivalent to doing the marginalization term by term (triangle by triangle) of equation (3.7) and multiplying them together. The idea can be illustrated by a simple example. For the graph in Figure 2.5, if A is missing, then the marginalized PDF $P(B, C, D, E)$ can be computed as in equation (3.8). In the case of C missing, if we assume that D is conditionally independent of A given B, and E is independent of A and B given D, which is a more demanding conditional independence requirement than that of equation (2.5), then,

$$P(A, B, D, E) = P(A, B) \cdot P(D|B) \cdot P(E|D) \tag{3.9}$$

In the case of D missing, if we assume that E is conditionally independent of A and B given C, which is also a more demanding conditional independence requirement than that of equation (2.5), then,

$$P(A, B, C, E) = P(A, B, C) \cdot 1 \cdot P(E|C) \tag{3.10}$$

Each term on the right-hand sides of equations (3.8), (3.9), and (3.10) is the marginalized version of its corresponding term in equation (2.5).

Similarly, under some stronger conditional independence, we can obtain an approximation of $P_{\overline{\mathcal{L}}_{body}}(\overline{X}_{body})$ by performing the marginalization term by term of equation (3.7). For example, considering triangle (A_t, B_t, C_t), $1 \leq t \leq T-1$, if all of A_t, B_t and C_t are present, then the tth term of equation (3.7) is $P_{A_t|B_t,C_t}(X_{A_t}|X_{B_t}, X_{C_t})$; if A_t is missing, the marginalized version of it is 1; if A_t and C_t are observed, but B_t is missing, it becomes $P_{A_t|C_t}(X_{A_t}|X_{C_t})$; if A_t exists but both B_t and C_t missing, it is $P_{A_t}(X_{A_t})$. The foreground probability $P_{\overline{\mathcal{L}}_{body}}(\overline{X}_{body})$ can be approximated by the product of the above (conditional) probability densities. Note that if too many body parts are missing, the conditional independence assumptions of the graphical model may no longer hold; it is reasonable to assume that the wrist is condition-

ally independent of the rest of the body given the shoulder and elbow, but if both shoulder and elbow are missing, this is no longer true. We will explore more on this issue later in this thesis. All the above (conditional) probability densities can be estimated from the training data. For instance, $P_{A_t|B_t,C_t}(X_{A_t}|X_{B_t}, X_{C_t})$ can be obtained via $P_{A_t,B_t,C_t}(X_{A_t}, X_{B_t}, X_{C_t})$ and $P_{B_t,C_t}(X_{B_t}, X_{C_t})$ as in equation (2.12), and $P_{A_t|C_t}(X_{A_t}|X_{C_t})$ can be computed through $P_{A_t,C_t}(X_{A_t}, X_{C_t})$ and $P_{C_t}(X_{C_t})$.

3.1.3 Comparison of two labelings and cost functions for dynamic programming

The best labeling ($\overline{L}^*$) can be found by comparing all the possible labelings. To compare two labelings $\overline{L}^1$ and $\overline{L}^2$, if we can assume the priors $P(\overline{L}^1)$ and $P(\overline{L}^2)$ are equal, then by equations (2.2) and (3.6),

$$\begin{aligned}
\frac{P(\overline{L}^1|\overline{X})}{P(\overline{L}^2|\overline{X})} &= \frac{P(\overline{X}|\overline{L}^1)}{P(\overline{X}|\overline{L}^2)} \\
&= \frac{P_{\overline{\mathcal{L}}^1_{body}}(\overline{X}^1_{body}) \cdot P_{bg}(\overline{X}^1_{bg})}{P_{\overline{\mathcal{L}}^2_{body}}(\overline{X}^2_{body}) \cdot P_{bg}(\overline{X}^2_{bg})} \\
&= \frac{P_{\overline{\mathcal{L}}^1_{body}}(\overline{X}^1_{body}) \cdot (1/S)^{N-K_1}}{P_{\overline{\mathcal{L}}^2_{body}}(\overline{X}^2_{body}) \cdot (1/S)^{N-K_2}} \\
&= \frac{P_{\overline{\mathcal{L}}^1_{body}}(\overline{X}^1_{body}) \cdot (1/S)^{M-K_1}}{P_{\overline{\mathcal{L}}^2_{body}}(\overline{X}^2_{body}) \cdot (1/S)^{M-K_2}}
\end{aligned} \tag{3.11}$$

where $\overline{\mathcal{L}}^1_{body}$ and $\overline{\mathcal{L}}^2_{body}$ are the sets of observed body parts for $\overline{L}^1$ and $\overline{L}^2$, respectively, K_1 and K_2 are the sizes of $\overline{\mathcal{L}}^1_{body}$ and $\overline{\mathcal{L}}^2_{body}$, and M is the total number of body parts ($M = 16$ here). $P_{\overline{\mathcal{L}}^i_{body}}(\overline{X}^i_{body})$, $i = 1, 2$, can be approximated as in section 3.1.2. From equation (3.11), the best labeling $\overline{L}^*$ is the $\overline{L}$ which can maximize $P_{\overline{\mathcal{L}}_{body}}(\overline{X}_{body}) \cdot (1/S)^{M-K}$. This formulation makes both search by dynamic programming and detection in different frames (possibly with different numbers of candidate features N) easy, as will be explained below.

At each stage of the dynamic programming algorithm described in section 2.3, the local optimum is stored according to the total value so far $T_t(X_{A_t}, X_{B_t}, X_{C_t})$, which is the sum of the local cost of the current triangle $\Psi_t(X_{A_t}, X_{B_t}, X_{C_t})$ and the costs of all the triangles on the path of the deletion of the current triangle. This requires that the local cost function $\Psi_t(X_{A_t}, X_{B_t}, X_{C_t})$ be comparable for different labelings: whether there are missing part(s) or not. Therefore we cannot only use the terms of $P_{\overline{\mathcal{L}}_{body}}(\overline{X}_{body})$, because, for example, as we discussed in the previous subsection, the tth term of $P_{\overline{\mathcal{L}}_{body}}(\overline{X}_{body})$ is $P_{A_t|B_t,C_t}(X_{A_t}|X_{B_t}X_{C_t})$ when all the three parts are present and it is 1 when A_t is missing. It is unfair to compare $P_{A_t|B_tC_t}(X_{A_t}|X_{B_t}, X_{C_t})$ with 1 directly. At this point, it is useful to notice that in $P_{\overline{\mathcal{L}}_{body}}(\overline{X}_{body}) \cdot (1/S)^{M-K}$, for each unobserved (missing) body part ($M-K$ in total), there is a $1/S$ term. $1/S$ (S is the volume of the space the position and velocity of a background point lies in) can be a reasonable local cost for a triangle with missing vertex A_t (the vertex to be deleted) because then for the same stage, the dimension of the domain of the local cost function is the same. Also, $1/S$ can be thought of as a threshold of $P_{A_t|B_tC_t}(X_{A_t}|X_{B_t}, X_{C_t})$, namely, if $P_{A_t|B_tC_t}(X_{A_t}|X_{B_t}, X_{C_t})$ is smaller than $1/S$, then the hypothesis that A_t is missing will win. Therefore, the local cost function $(\exp(\Psi_t(X_{A_t}, X_{B_t}, X_{C_t})))$ for the tth $(1 \leq t \leq T-1)$ triangle can be approximated as follows:

- if all the three body parts are observed, it is $P_{A_t|B_t,C_t}(X_{A_t}|X_{B_t}, X_{C_t})$;
- if A_t is missing or two or three of A_t, B_t, C_t are missing, it is $1/S$;
- if B_t or C_t is missing and the other two body parts are observed, it is $P_{A_t|C_t}(X_{A_t}|X_{C_t})$ or $P_{A_t|B_t}(X_{A_t}|X_{B_t})$.

The same idea can be applied to the last triangle T. These approximations are to be validated in experiments. Notice that when two body parts in a triangle are missing, only velocity information for the third body part is available since we use relative positions. The velocity of a point alone does not have much information, so for two parts missing, we use the same cost function as the case of three body parts missing.

The approximation of the local cost functions described above can be illustrated by a simple example of Figure 2.5 (with $M = 5$). We want to compare a labeling $\overline{L}^1$ with all five vertices (A, B, C, D, E) present and another labeling $\overline{L}^2$ with D missing.

By equations (2.5), (3.10) and (3.11), we need to compute

$$\begin{aligned} & \frac{P(A,B,C,D,E)}{P(A,B,C,E)\cdot(1/S)} \\ = \; & \frac{P(A,B,C)\cdot P(D|B,C)\cdot P(E|C,D)}{P(A,B,C)\cdot 1\cdot P(E|C)\cdot(1/S)} \\ = \; & \frac{P(A,B,C)}{P(A,B,C)}\cdot\frac{P(D|B,C)}{(1/S)}\cdot\frac{P(E|C,D)}{P(E|C)} \end{aligned} \tag{3.12}$$

The last line of equation (3.12) gives the local cost for each triangle.

With the local cost functions defined above, dynamic programming can be used to find the labeling with the highest $P_{\overline{\mathcal{L}}_{body}}(\overline{X}_{body})\cdot(1/S)^{M-K}$. The computational complexity is on the order of $M * N^3$.

3.2 Detection

In the previous section, we explain how to compute the likelihood of a hypothesis labeling $\overline{L}$, $P(\overline{X}|\overline{L})$, and how to compare two labelings and obtain the best labeling. Based on these tools, we are now ready to discuss how detection is performed. Let O_1 denote a person present in the image, and O_0 absent. The detection task is to determine whether the ratio

$$\begin{aligned} \frac{P(O_1|\overline{X})}{P(O_0|\overline{X})} &= \frac{P(\overline{X}|O_1)P(O_1)/P(\overline{X})}{P(\overline{X}|O_0)P(O_0)/P(\overline{X})} \\ &= \frac{P(\overline{X}|O_1)}{P(\overline{X}|O_0)}\cdot\frac{P(O_1)}{P(O_0)} \end{aligned} \tag{3.13}$$

is greater than 1. If we assume the priors are equal, the second term of the above equation is 1. We need to compute $P(\overline{X}|O_1)$ and $P(\overline{X}|O_0)$. Assume $\mathcal{L}$ is the set of all the possible labelings when a person present (O_1), then,

$$\begin{aligned} P(\overline{X}|O_1) &= \sum_{\overline{L}\in\mathcal{L}} P(\overline{X},\overline{L}|O_1) \\ &= \sum_{\overline{L}\in\mathcal{L}} P(\overline{X}|\overline{L},O_1)P(\overline{L}|O_1) \end{aligned} \tag{3.14}$$

When there is no person in the scene, the only possible labeling is $\overline{L}_0 = [BG, BG, \ldots, BG]$. Then,

$$\begin{aligned} P(\overline{X}|O_0) &= P(\overline{X},\overline{L}_0|O_0) \\ &= P(\overline{X}|\overline{L}_0,O_0)P(\overline{L}_0|O_0) \\ &= P(\overline{X}|\overline{L}_0,O_0) \end{aligned} \tag{3.15}$$

If we assume the position and velocity of the visible body parts are independent of position and velocity of background features (clutter) and the background features are independently uniformly distributed, then

$$P(\overline{X}|\overline{L},O_1) = P_{\overline{\mathcal{L}}_{body}}(\overline{X}_{body}) \cdot P_{bg}(\overline{X}_{bg}) = P_{\overline{\mathcal{L}}_{body}}(\overline{X}_{body}) \cdot (1/S)^{N-K_{\overline{L}}} \tag{3.16}$$

$$P(\overline{X}|\overline{L}_0,O_0) = P_{bg}(\overline{X}) = (1/S)^N \tag{3.17}$$

where we use the same notation as in section 3.1: N is the number of candidate features, $N-K_{\overline{L}}$ is he number of background points for labeling $\overline{L}$, and S is the volume of the space X_i can be in. Under the independent uniform background assumption, part of the background terms in $P(\overline{X}|\overline{L},O_1)$ and $P(\overline{X}|\overline{L}_0,O_0)$ can be canceled out (similar to the last equal sign of equation (3.11)). Substituting equations (3.14) to (3.17) into equation (3.13), we get

$$\frac{P(O_1|\overline{X})}{P(O_0|\overline{X})} = \frac{\sum_{\overline{L}\in\mathcal{L}} P_{\overline{\mathcal{L}}_{body}}(\overline{X}_{body}) \cdot (1/S)^{M-K_{\overline{L}}} \cdot P(\overline{L}|O_1)}{(1/S)^M} \cdot \frac{P(O_1)}{P(O_0)} \tag{3.18}$$

where M is the number of body parts, a fixed number across all the frames and all the hypothesis labelings, and $P(O_1)$ and $P(O_0)$ do not depend on $\overline{X}$ either. Therefore, detection can be performed by thresholding

$$\sum_{\overline{L}\in\mathcal{L}} P_{\overline{\mathcal{L}}_{body}}(\overline{X}_{body}) \cdot (1/S)^{M-K_{\overline{L}}} \cdot P(\overline{L}|O_1) \tag{3.19}$$

without accurately estimating background probabilities and prior probabilities $P(O_1)$ and $P(O_0)$.

In equation (3.19), $P_{\overline{\mathcal{L}}_{body}}(\overline{X}_{body}) \cdot (1/S)^{M-K_{\overline{L}}}$ can be computed as in section 3.1. $P(\overline{L}|O_1)$ can be estimated by the following two strategies: one is 'winner-take-all,' and the other is to assume that all the labelings are equally likely.

3.2.1 Winner-take-all

From section 3.1, the labeling $\overline{L}$ with the highest $P_{\overline{\mathcal{L}}_{body}}(\overline{X}_{body}) \cdot (1/S)^{M-K_{\overline{L}}}$ provides us with the most human-like configuration out of all the candidate labelings. We call it the best labeling $\overline{L}^*$. In the winner-take-all strategy, we take $P(\overline{L}^*|O_1) = 1$ and $P(\overline{L}|O_1) = 0$ for other labeling $\overline{L}$'s in equation (3.19). Therefore detection is done by thresholding the likelihood of the best labeling $\overline{L}^*$, $P_{\overline{\mathcal{L}}^*_{body}}(\overline{X}_{body}) \cdot (1/S)^{M-K_{\overline{L}^*}}$. The threshold needs to be set based on experiments to ensure the best trade-off between false acceptance and false rejection errors.

3.2.2 Summation over all the hypothesis labelings

Another simple and reasonable strategy is to assume that all the hypothesis labelings are equally likely, that is, for any labeling $\overline{L}$, $P(\overline{L}|O_1) = 1/|\mathcal{L}|$, where $|\mathcal{L}|$ is the number of possible labelings. Under this assumption, formula (3.19) becomes $(1/|\mathcal{L}|) \cdot \sum_{\overline{L}\in\mathcal{L}} P_{\overline{\mathcal{L}}_{body}}(\overline{X}_{body}) \cdot (1/S)^{M-K_{\overline{L}}}$. It is computationally prohibitive to perform the summation in a brute-force way. Fortunately, the probability decomposition allows us to do the summation efficiently, as will be explained below.

We first consider the problem where there are no missing body parts if a person is

present. In this case, detection depends on $(1/|\mathcal{L}|) \cdot \sum_{\overline{L} \in \mathcal{L}} P_{\mathcal{S}_{body}}(\overline{X}_{body})$. By equation (2.8),

$$\begin{aligned} & P_{\mathcal{S}_{body}}(X_{S1}, X_{S2}, \ldots X_{SM}) \\ = & \prod_{t=1}^{T-1} P_{A_t|B_tC_t}(X_{A_t}|X_{B_t}, X_{C_t}) \cdot P_{A_TB_TC_T}(X_{A_T}, X_{B_T}, X_{C_T}) \end{aligned}$$

then,

$$\begin{aligned} & \sum_{\overline{L} \in \mathcal{L}} P_{\mathcal{S}_{body}}(\overline{X}_{body}) \\ = & \sum_{\overline{L} \in \mathcal{L}} \prod_{t=1}^{T-1} P_t(X_{A_t}|X_{B_t}, X_{C_t}) P_T(X_{A_T}, X_{B_T}, X_{C_T}) \\ = & \sum_{X_{A_T}, X_{B_T}, X_{C_T}} P_T(X_{A_T}, X_{B_T}, X_{C_T}) \sum_{X_{A_{T-1}}} \cdots \sum_{X_{A_2}} P_2(X_{A_2}|X_{B_2}, X_{C_2}) \sum_{X_{A_1}} P_1(X_{A_1}|X_{B_1}, X_{C_1}) \end{aligned} \tag{3.20}$$

where the summation $(\sum)$ is conducted from right to left. Comparing with equation (2.10), we can see that the only difference is that here the operation is 'sum' instead of 'max'. Therefore, if we replace the 'max' operation with 'sum' operation, the dynamic programming procedure described in section 2.3 can be applied to compute the summation. Let

$$\Psi_t(X_{A_t}, X_{B_t}, X_{C_t}) = P_{A_t|B_tC_t}(X_{A_t}|X_{B_t}, X_{C_t}), \text{ for } 1 \leq t \leq T-1 \tag{3.21}$$

$$\Psi_t(X_{A_t}, X_{B_t}, X_{C_t}) = P_{A_TB_TC_T}(X_{A_T}, X_{B_T}, X_{C_T}), \text{ for } t = T \tag{3.22}$$

be the cost function* associated with each triangle, then the summation can be performed as follows:

*In section 2.3, we use $\log P_t(X_{A_t}|X_{B_t}, X_{C_t})$ as cost function for numerical reasons (a value from a high dimensional Gaussian distribution can be very small). But here it is the summation of probabilities, so it is hard to use log directly. The trick we used to avoid underflow is to compute $P_t \cdot S = exp(log(P_t) + log(S))$, where S is the volume of the uniform background.

Stage 1: for every pair (X_{B_1}, X_{C_1}),

Compute $\Psi_1(X_{A_1}, X_{B_1}, X_{C_1})$ for all possible X_{A_1}

Define $T_1(X_{A_1}, X_{B_1}, X_{C_1})$ the total value so far.

Let $T_1(X_{A_1}, X_{B_1}, X_{C_1}) = \Psi_1(X_{A_1}, X_{B_1}, X_{C_1})$

Store $\Gamma_1(X_{B_1}, X_{C_1}) = \sum_{X_{A_1}} T_1(X_{A_1}, X_{B_1}, X_{C_1})$

Stage t, $2 \leq t \leq T$: for every pair (X_{B_t}, X_{C_t}),

Compute $\Psi_t(X_{A_t}, X_{B_t}, X_{C_t})$ for all possible X_{A_t}

Compute the total value so far (till stage t):

– Define $T_t(X_{A_t}, X_{B_t}, X_{C_t})$ the total value so far.

Initialize $T_t(X_{A_t}, X_{B_t}, X_{C_t}) = \Psi_t(X_{A_t}, X_{B_t}, X_{C_t})$

– If edge (A_t, B_t) is contained in a previous

stage and τ is the latest such stage, multiply

$\Gamma_\tau(X_{A_t}, X_{B_t})$ (or $\Gamma_\tau(X_{B_t}, X_{A_t})$ if the edge

was reversed) to $T_t(X_{A_t}, X_{B_t}, X_{C_t})$

– Likewise, multiply the values of the latest previous

stages containing respectively edge (A_t, C_t) and edge

(B_t, C_t) to $T_t(X_{A_t}, X_{B_t}, X_{C_t})$

Store $\Gamma_t(X_{B_t}, X_{C_t}) = \sum_{X_{A_t}} T_t(X_{A_t}, X_{B_t}, X_{C_t})$

When stage T calculation is complete, the overall sum can be obtained by

$$\sum_{\overline{L} \in \mathcal{L}} P_{\mathcal{S}_{body}}(\overline{X}_{body}) = \sum_{X_{B_T}, X_{C_T}} \Gamma_T(X_{B_T}, X_{C_T}) \tag{3.23}$$

The computational complexity of the above method is on the order of $M * N^3$.

When there is occlusion, the local cost function $\Psi_t(X_{A_t}, X_{B_t}, X_{C_t})$ associated with each triangle t, $(1 \leq t \leq T)$, can be approximated as in section 3.1.3.

3.2.3 Discussion

In the above sections we present two ways to estimate the prior probabilities of different labelings $P(\overline{L}|O_1)$, and therefore two ways to do detection. From the performance point of view, which estimation is better depends on which one gives a closer approximation to the 'truth'. If the best labeling is with much higher likelihood than other labelings, the winner-take-all strategy is more 'correct'. If there are some labelings with similar likelihoods, then the summation of all possible labelings works better. From the computational point of view, winner-take-all strategy is more efficient because it can do detection and labeling at the same time. For sum-over-all-labelings strategy, extra computation is needed to obtain localization and labeling.

There are also other ways to model $P(\overline{L}|O_1)$. For instance, if we have some prior knowledge on the number of background (clutter) points, $P(\overline{L}|O_1)$ can be more precisely estimated. In [22], 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 described in sections 2.3 and 3.2.2.

In this chapter we describe the labeling problem (section 3.1) before the detection problem (section 3.2). This is a convenient way to explain things because section 3.1 provides tools for section 3.2. However, in application, we run detection first to decide whether there is a person in the scene, and then labeling if necessary.

3.3 Integrating temporal information

So far, we have only assumed that we may use information from two consecutive frames, from which we obtain position and velocity of a number of features. In this section we extend our previous results to the case where multiple frames are available. However, in order to maintain generality we will assume that tracking across more than two frames is impossible and therefore that the measurements from one pair of

frames to the next are uncorrelated. This is a simplified model of the situation where, due to extreme body motion or to loose and textured clothing and occlusion, tracking is extremely unreliable and each feature's lifetime is short. Neri et al. [23] used similar assumption when conducting their psychophysical investigation of biological motion perception in the human visual system.

Let $P(O|\overline{X})$ denote the probability of the existence of a person given $\overline{X}$ observed. From section 3.2, we use the approximation: $P(O|\overline{X})$ is proportional to $\Phi(\overline{X})$, which is defined as

$$\Phi(\overline{X}) \stackrel{\text{def}}{=} \begin{cases} \max_{\overline{L}\in\mathcal{L}} P_{\overline{\mathcal{L}}_{body}}(\overline{X}_{body}) \cdot (1/S)^{M-K_{\overline{L}}} & \text{if winner-take-all} \\ \frac{1}{|\mathcal{L}|} \cdot \sum_{\overline{L}\in\mathcal{L}} P_{\overline{\mathcal{L}}_{body}}(\overline{X}_{body}) \cdot (1/S)^{M-K_{\overline{L}}} & \text{if sum-over-all-labelings} \end{cases} \tag{3.24}$$

Now if we have n observations $\overline{X}_1, \overline{X}_2, \ldots, \overline{X}_n$, the decision depends on

$$\begin{aligned} & P(O|\overline{X}_1, \overline{X}_2, \ldots, \overline{X}_n) \\ = \; & \frac{P(\overline{X}_1, \overline{X}_2, \ldots, \overline{X}_n|O) \cdot P(O)}{P(\overline{X}_1, \overline{X}_2, \ldots, \overline{X}_n)} \\ = \; & \frac{P(\overline{X}_1|O)P(\overline{X}_2|O)\ldots P(\overline{X}_n|O) \cdot P(O)}{P(\overline{X}_1, \overline{X}_2, \ldots, \overline{X}_n)} \end{aligned} \tag{3.25}$$

The last line of the above equation holds if we assume that $\overline{X}_1, \overline{X}_2, \ldots, \overline{X}_n$ are independent observations. Assuming the priors are equal, $P(O|\overline{X}_1, \overline{X}_2, \ldots, \overline{X}_n)$ can be represented by $P(\overline{X}_1|O)P(\overline{X}_2|O)\ldots P(\overline{X}_n|O)$, which is proportional to $\prod_{i=1}^{n}\Phi(\overline{X}_i)$. Each $\Phi(\overline{X}_i)$ can be evaluated as equation (3.24). If we set up a threshold for $\prod_{i=1}^{n}\Phi(\overline{X}_i)$, we can do detection given $\overline{X}_1, \overline{X}_2, \ldots, \overline{X}_n$.

3.4 Counting

Counting how many people are in the scene is also an important task since images often have multiple people in them. By the method described above, we can first detect whether a person is present. If so, all the points belonging to the most human-like

configuration are deleted and the next detection and labeling can then be performed from the remaining points. We can keep doing this until no person is detected.

Assume M is the number of body parts, N is the number of candidate markers, and P is the number of people in the scene. The cost of detecting the first person is on the order of $M * N^3$, the cost for the second person is of $M * (N - m_1)^3$, where m_1, $(m_1 \leq M)$, is the number of body parts present in the first person, and so on. Therefore, the total cost of counting P individuals is of $P * M * N^3$.

3.5 Experiments on motion capture data

In this section experiments are conducted on motion capture data (as in section 2.4) with occlusion and added clutter. We test and compare the two detection strategies and analyze the detection and labeling rates as functions of the number of visible body parts, with and without integration of temporal information. We also analyze the performance of estimating the number of people in the scene.

3.5.1 Detection and labeling

Detection is to distinguish whether or not a person is present in the scene (Figure 3.2). In this experiment, we first test and compare two detection strategies described in section 3.2. We present the algorithm with two types of inputs (presented randomly in equal proportions); in one case only clutter (background) points are present, in the other a pre-determined number of randomly selected body parts in the set of test data are superimposed on some clutter. If there are body parts in the scene and the program thinks there is a person, the person is correctly detected. If there are only background points in the scene but the program thinks there is a person, a false alarm happens. We measure the frequency of correct detections and false alarms, and build receiver operating characteristics (ROC) curves for our detector.

We want to test the detection performance when only part of the whole body (with 16 body parts in total) can be seen. We generated the signal points (body parts) in a frame in the following way: for a fixed number of signal points, we randomly selected

which body parts to be used for each frame (actually pair of frames, since consecutive frames are used to estimate the velocity of each body part). Therefore in principle, each body part has an equal chance to be represented, and as far as the decomposed body graph is concerned, all kinds of graph structures (with different body parts missing) can be tested.

The positions and velocities of clutter (background) points were independently generated from uniform distributions of their corresponding ranges. For positions, we used the leftmost and rightmost positions of the whole sequence as its horizontal range, and highest and lowest body part positions as its vertical range. For velocities, the possible range is inside a circle of the velocity space (horizontal and vertical velocities) with radius of the maximum magnitude of the velocities from the real sequences. Figure 3.2 (a) shows a frame with 8 body parts and 30 added background points with arrows representing velocities.

Figure 3.3 shows the experimental results on walking sequences (sequences W3 and W4, sequence W3 was used for the training and W4 for testing). Figure 3.3(a) is from the winner-take-all detection strategy. The six solid curves show the receiver operating characteristics (ROCs) of 3 to 8 signal points with 30 added background points vs. 30 background points. The bigger the number of signal points observed, the better the ROC is. With 30 background points, when the number of signal points is more than 8, the ROCs are almost perfect.

In practice, when using the detector, some detection threshold needs to be set. If the likelihood exceeds the threshold, a person is deemed to be present. Since the number of body parts is unknown before detection, we need to fix a threshold that is independent of (and robust with respect to) the number of body parts present in the scene. The dashed line in Figure 3.3 (a) shows the overall ROC of all the frames used for the six ROC curves in solid lines. We took the threshold when $P_{detect} = 1 - P_{false-alarm}$ on that curve as our threshold. The star ('*') point on each solid curve shows the point corresponding to the threshold.

Figure 3.3 (b) shows the ROC curves from the sum-over-all-labelings strategy. The experiment settings are the same as (a), except a different detection algorithm

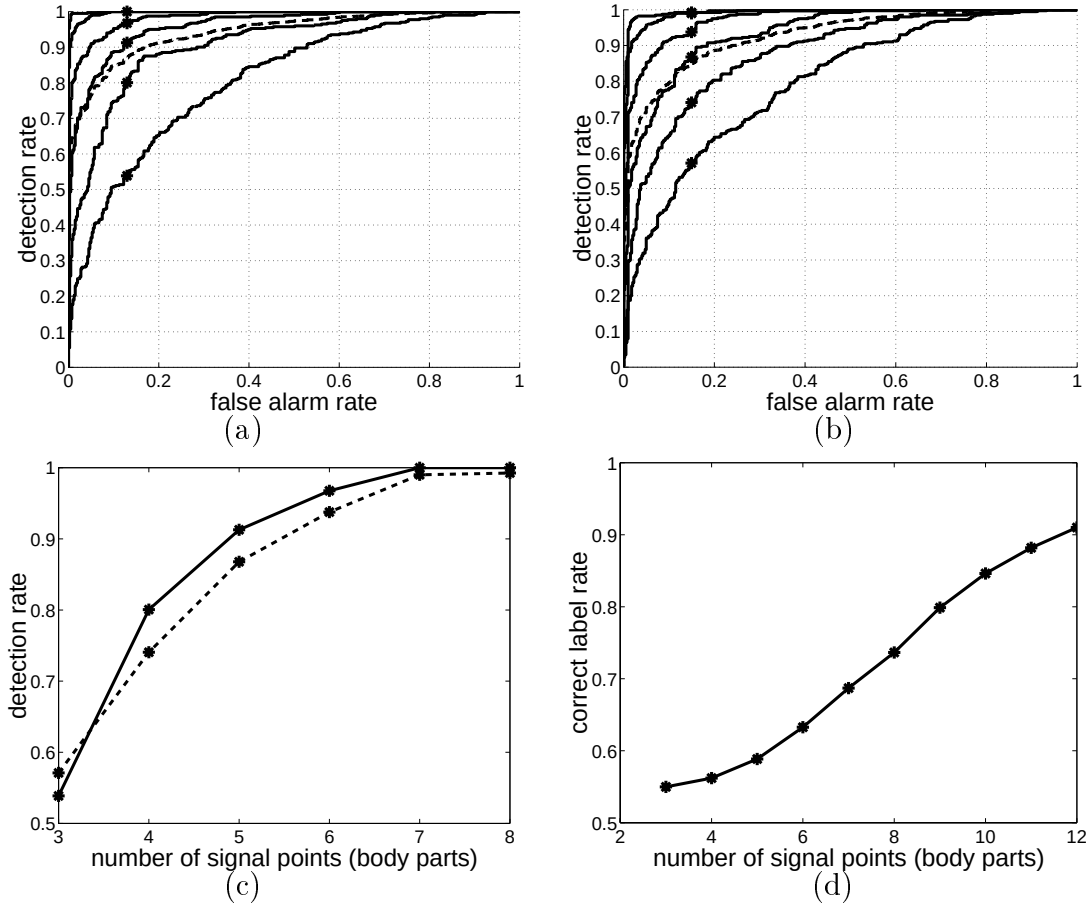


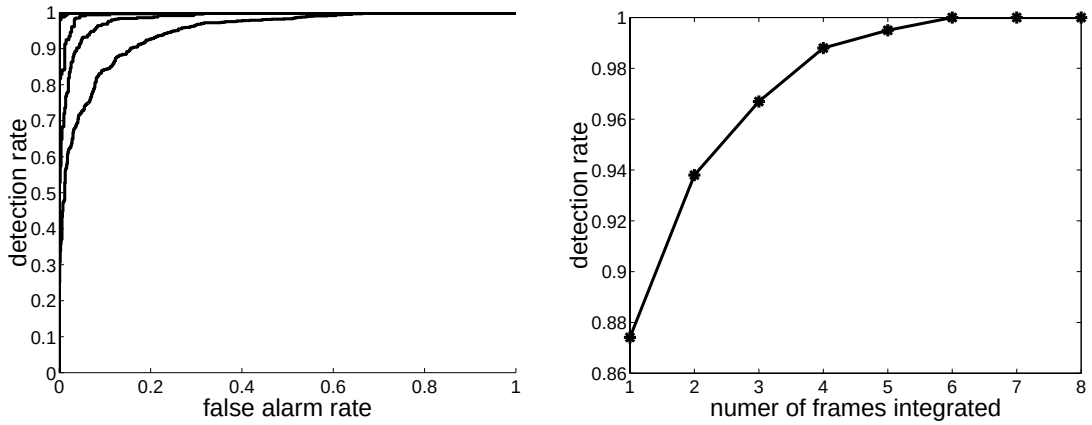
Figure 3.3: Detection and labeling results on motion capture data (under the conditions of clutter and occlusion). **(a)** ROC curves from the winner-take-all detection strategy. Solid lines: 3 to 8 body parts with 30 background points vs. 30 background points only. The bigger the number of signal points is, the better the ROC is; dashed line: overall ROC considering all the frames used in six solid ROCs. The stars ('*') on the solid curves are the points corresponding to the threshold where $P_D = 1 - P_{FA}$ on the dashed overall ROC curve. **(b)** ROC curves from the sum-over-all-labelings strategy. The experiment settings are the same as (a), except a different detection algorithm is used. **(c)** detection rate vs. number of body parts displayed. Solid line: from the winner-take-all strategy with regard to the fixed threshold where $P_D = 1 - P_{FA}$ on the overall ROC curve in (a), with false alarm rate $P_{FA} = 12.97\%$; dashed line: from the sum-over-all-labelings strategy with regard to the fixed threshold where $P_D = 1 - P_{FA}$ on the overall ROC curve in (b), with $P_{FA} = 14.96\%$. **(d)** correct label rate (label-by-label rate) vs. number of body parts when a person is correctly detected (using the winner-take-all strategy with regard to the same threshold as in (c)).

is used. Figure 3.3 (c) shows the relation between detection rate and the number of body parts displayed. The solid line is from the winner-take-all detection strategy with regard to the fixed threshold where $P_D = 1 - P_{FA}$ on the overall ROC curve in Figure 3.3 (a), with false alarm rate $P_{FA} = 12.97\%$; and the dashed line is from the sum-over-all-labelings detection strategy with regard to the fixed threshold where $P_D = 1 - P_{FA}$ on the overall ROC curve in (b), with $P_{FA} = 14.96\%$. From Figure 3.3, we can see that both detection algorithms can work well: even when only three body parts are present in the scene, the detection performance is much better than the chance level. However, the winner-take-all strategy works better than the sum-over-all-labelings strategy for these data. This is reasonable because for motion capture data, one joint is represented by one dot, and therefore, there is only one correct labeling (or a very small number of close-to-correct labelings), which is much better than other labelings. This is in contrast to the situation of gray-scale images where there can be many close candidate features for one body part and therefore many labelings may be comparable. Since the winner-take-all strategy works better on motion capture data, we will use it as the detection method in the later experiments on motion capture data.

When the algorithm can correctly detect whether a person is there, it does not necessarily mean that all the body parts are correctly labeled. We studied the correct label rate (*label-by-label rate*) when a person is correctly detected. An error happens when a body part is assigned a wrong candidate feature. Figure 3.3 (d) shows the result. While the detection rate keeps constant (almost 1) with 8 or more body parts visible, the correct label rate goes up as the number of body parts increases. The correct label rates here are smaller than the results in section 2.4 since we have less signal points but many more background points. If the average number of features detected is N, (N is more than 30 in this experiment), the chance level of a body part being assigned a correct candidate feature by random selection is $1/(N+1)$ (with one more background point). The correct rate here is much higher than that, more than 50% with only 3 body parts (almost 20 times above chance level) and exceeding 90% when 12 out of the 16 body parts are present.

3.5.2 Using temporal information

Here we tested how the detection rate improved by integrating information over time, using the approach described in section 3.3. We used the data of 5 signal points and 30 background points in each frame to test the performance of using information from multiple frames (the body parts present in each frame were chosen randomly and independently). Figure 3.4 (a) shows ROC curves of using n ($n = 1, ...8$) frames. The bigger n is, the better the ROC curve is. When $n > 5$, the ROCs are almost perfect and overlapped with the axes. If Θ is the likelihood threshold of $P_{detect} = 1 - P_{false-alarm}$ when only one frame is used, then the threshold of $P_{detect} = 1 - P_{false-alarm}$ for using n frames is Θ^n. Figure 3.4 (b) plots the detection rate (with $P_{detect} = 1 - P_{false-alarm}$) vs. the number of frames integrated. The results get better with more frames used, and even with only 5 body parts present it is possible to get completely accurate detection after combining information from only 6 frames.



(a) (b)

Figure 3.4: Results of integrating multiple frames. **(a)** ROCs of integrating one to eight frames using only 5 body parts with 30 clutter points present. The more frames integrated, the better the ROC curve is. When more than five frames are used, the ROCs are almost perfect and overlapped with the axes. **(b)** detection rate (when $P_{detect} = 1 - P_{false-alarm}$) vs. number of frames used.

3.5.3 Counting experiments

The counting task is to find how many people are in a scene given a number of observed points (with position and velocity). A person was generated by randomly choosing a frame from the sequence, and several frames (persons) can be superimposed together into one image. In one image, the position of each person was randomly selected, but made sure not to overlap with each other. The background points were generated in a similar way to the detection and labeling experiments in section 3.5.1, but with the positions of the background points uniformly distributed on a window which is three times as wide as the window in Figure 3.2 (a). Figure 3.5 gives an example of images used in this experiment, with three persons (six body parts each) and sixty background points.

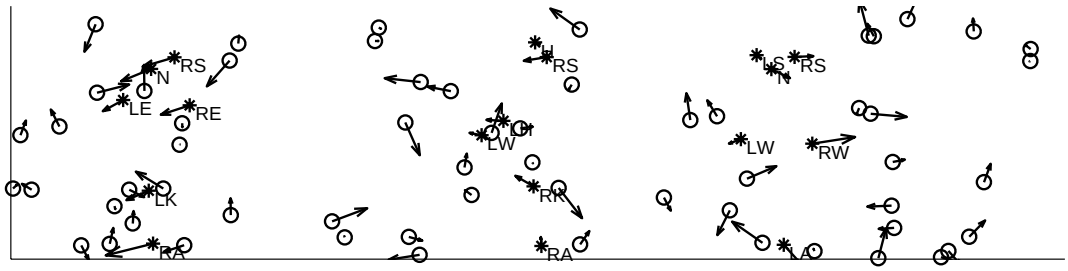


Figure 3.5: One sample image of counting experiments. '*' denotes body parts from a person and 'o's are background points. There are three persons (six body parts for each person) with sixty superimposed background points. Arrows are the velocities.

We did experiments on up to three persons in one image. We used the threshold from Figure 3.3(a). If the likelihood of the configuration found was above the threshold, then it was counted as a person. If the number of detected people provided by the algorithm was different (either more or less) from the ground truth, an error happened. The curves in Figure 3.6 show the correct rate vs. the number of signal points (body parts displayed) for each person. To compare the results conveniently, we used the same number of body parts for different persons in one image (but the parts appearing were randomly chosen). The solid line with stars is the result of one person in an image, the dashed line with circles is for two persons, and the dash-dot

line with triangles is for three persons. If there was no person in the image, the correct rate is 95%. From Figure 3.6, we see that the result for less people in an image is better than that of more people, especially when the number of body parts present is small. We can explain it as follows. If the probability of counting one person correctly is P, then the probability of counting n people correctly is P^n if the detection of different people is independent. For example, in the case of four body parts, for one person the correct rate is 0.6, then the correct rate for counting three person is $0.6^3 = 0.216$. But since we randomly chose the position of each person, body parts from different persons may be very close, so the independence couldn't be strictly held. Furthermore, the assumption of independence is also violated since once a person is detected, the corresponding body parts are removed from the scene in order to detect subsequent people.

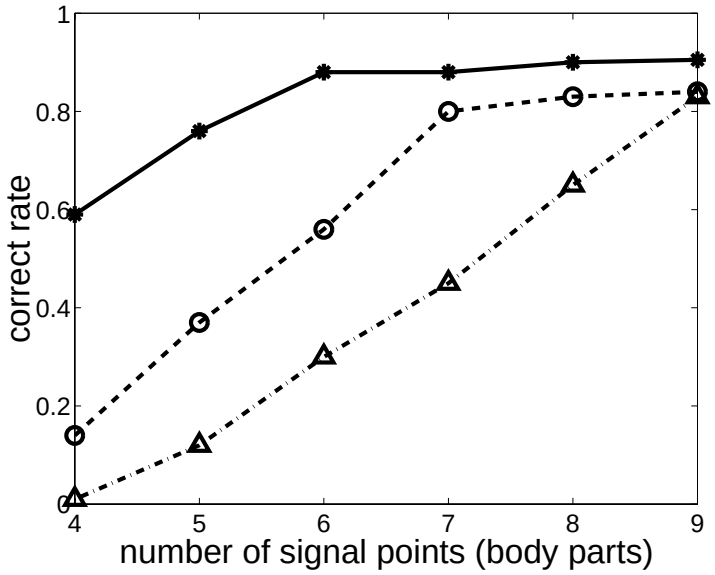
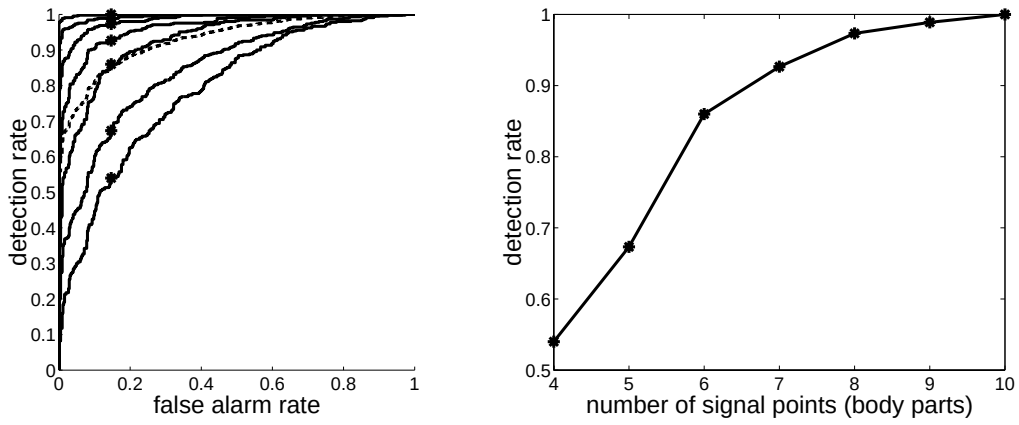


Figure 3.6: Results of counting people. Solid line (with *): one person; dashed line (with o): two persons; dash-dot line (with triangles): three persons. Counting is done with regard to the threshold chosen from Figure 3.3 (a). For that threshold the correct rate for recognizing that there is no person in the scene is 95%.



(a) (b)

Figure 3.7: Results of dancing sequences. (a) Solid lines: ROC curves for 4 to 10 body parts with 30 added background points vs. 30 background points only. The bigger the number of signal points is, the better the ROC is. Dashed line: overall ROC considering all the frames used in seven solid ROCs. The threshold corresponding to $P_D = 1 - P_{FA}$ on this curve was used for (b). The stars ('*') on the solid curves are the points corresponding to that threshold. (b) detection rate vs. the number of body parts displayed with regard to a fixed threshold at which $P_D = 1 - P_{FA}$ on the overall ROC curve in (a). The false alarm rate is 14.67%.

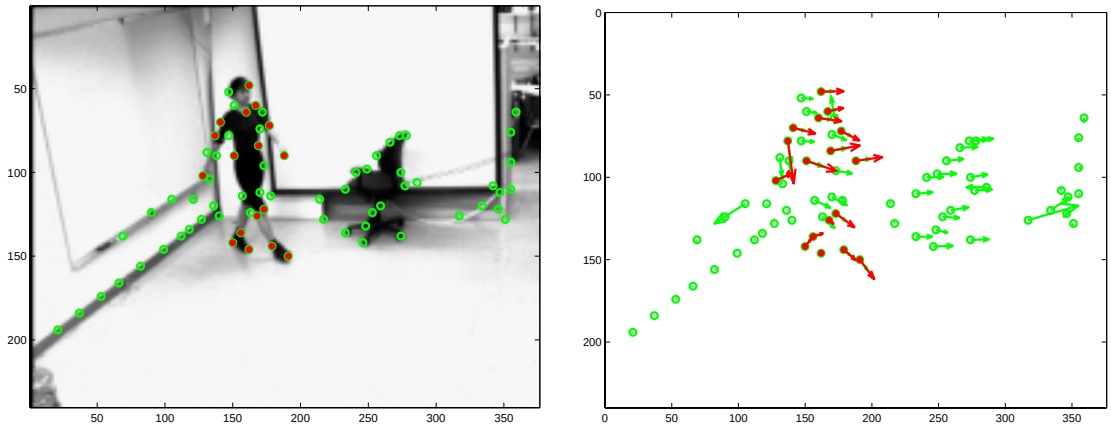
3.5.4 Experiments on dancing sequence

In this section, we performed detection experiments on the dancing sequence DA (the first half was used for training and the second half for testing). The seven solid curves of Figure 3.7 (a) are the ROC curves of 4 to 10 signal points with 30 added background points. The signal points are from the dancing sequence and the background points were generated the same way as in the detection and labeling experiments in section 3.5. In Figure 3.7 (a), the bigger the number of signal points observed, the better the ROC. The dashed line in Figure 3.7 (a) shows the overall ROC of all the frames used for the seven ROC curves in solid line. We take the threshold when $P_{detect} = 1 - P_{falsealarm}$ on that curve as our threshold and get the plot of detection rate vs. the number of signal points in Figure 3.7 (b). The false alarm rate is 14.67%. With more than 9 (out of 16) body parts present, the detection rate is almost 1. Comparing with the results in Figure 3.3, we can see that more body parts must be observed during the dancing sequence to achieve the same detection

rate as with the walking sequences, which is expected since the motion of dancing sequences is more active and harder to model. Nevertheless, the ROC curve with 10 out of 16 body parts present is nearly perfect.

3.6 Experiments on gray-scale image sequences

In this section, we conduct experiments on more challenging data: gray-scale image sequences. To apply the detection and labeling algorithms, candidate features are obtained from the Lucas-Tomasi-Kanade [1, 2] feature selector/tracker on pairs of frames. Figure 3.8 illustrates the approach.



(a) (b)

Figure 3.8: Illustration of the approach on gray-scale images. For a given image (a), features are first selected and tracked to the next frame. Dots in (a) are the features, and (b) shows the features with velocities. From all the candidate feature points (with positions and velocities), we want to first decide whether there is a person in the scene and then find the best labeling – the most human-like configuration (dark dots in (a) and (b)) according to a learned probabilistic model.

Figure 3.9 shows the hand-constructed probabilistic decomposition for the experiments. Twenty parts are chosen to represent the human body. The dark dots in Figure 3.8 shows features representing the parts. Three parts are missing for the frame in Figure 3.8: two at the left knee and one at the right heel.

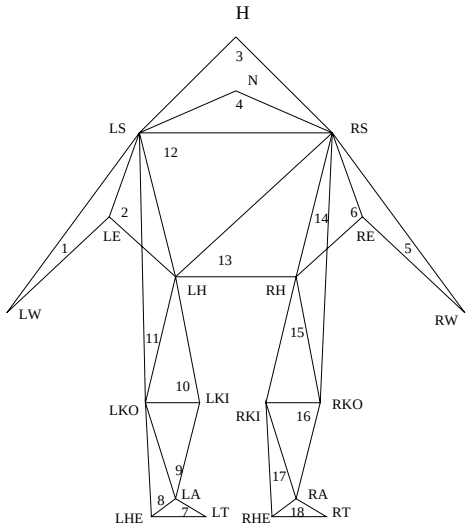


Figure 3.9: Decompositions of the human body for gray-scale image experiments. 'L' and 'R' in label names indicate left and right. H:head, N:neck, S:shoulder, E:elbow, W:wrist, H:hip, KI:inside knee, KO:outside knee, A:ankle, HE:heel, and T:toe. The numbers inside triangles give one elimination order.

3.6.1 Data

The image sequences were captured by a CCD camera at 30 Hz. There are three types of motion: (1) A subject walks from left to right, facing 60 degrees away from the front view (middle row of Figure 3.10). We have 20 sequences with around 120 frames each. (2) A chair moves from left to right (bottom row of Figure 3.10). 8 sequences, with 120 frames each. (3) While a subject walks as in type (1), a chair also moves as in type (2) (top row of Figure 3.10). 16 sequences, with 120 frames each.

Training set: manually tracked data. The model parameters (mean and covariance of Gaussian) are learned from a training set with the hand-constructed ground truth labeling. The training sequences include eight type (1) walking sequences. For the first frame of each sequence, we manually select all the features corresponding to the body parts in the model of Figure 3.9. The features are then tracked automatically to the next frame using the Lucas-Tomasi-Kanade tracking algorithm. The tracking results are monitored, and features with obvious tracking errors are discarded. The tracking procedure provides us with the positions and

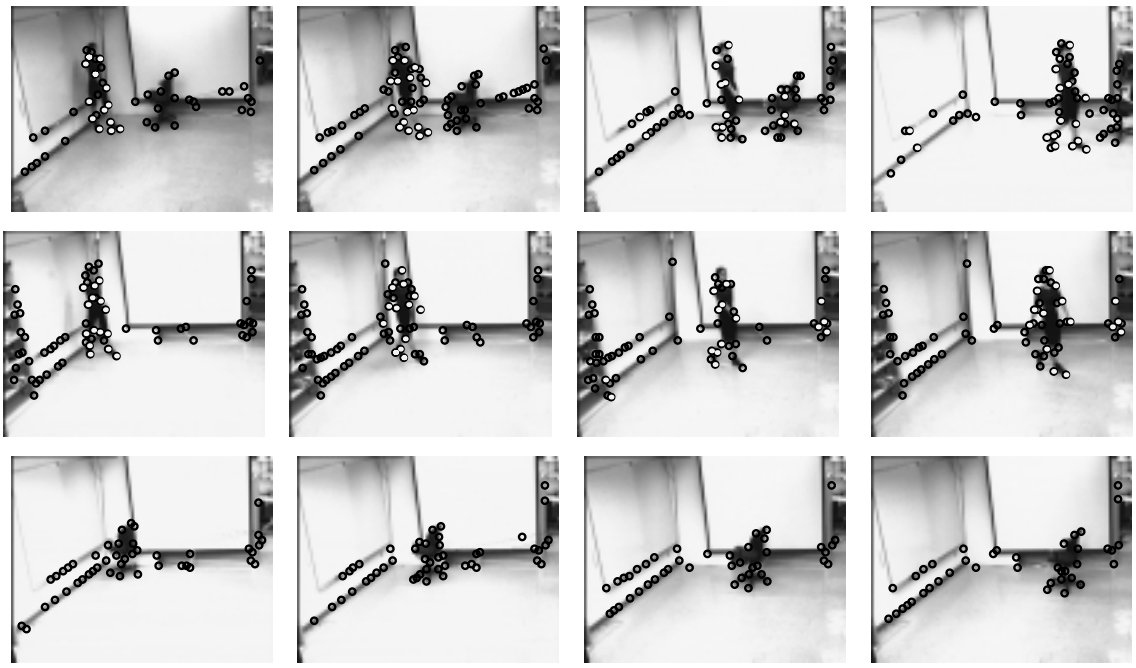


Figure 3.10: Sample frames from body and chair moving sequences (type (3), top row), body moving sequences (type (1), middle row), and chair moving sequences (type (2), bottom row). The dots (either in black or in white) are the features selected by Lucas-Tomasi-Kanade [1, 2] algorithm on pairs of frames. The white dots are the most human-like configuration found by our algorithm.

velocities of features. The labeling (body part assignment of the features) is given manually. This process is repeated for all the frames.

Testing Set. For the test sequences, features are obtained automatically from the standard Lucas-Tomasi-Kanade feature selection/tracking algorithm on pairs of frames. We do not track features over more than two frames, but reselect all the features at the next frame after tracking, which simulates the arguably most difficult situation for labeling and detection (as discussed in section 3.3). The dots in Figures 3.8 and 3.10 are features from this procedure. The average number of features detected in each frame is 64, 46, and 58 for type (1), (2), and (3) sequences, respectively. There are more body parts missing (occlusion) in the automatic detected features than in the manually tracked training data.

3.6.2 Labeling on manually tracked data

To evaluate the hand-crafted decomposable triangulated probabilistic model (Figure 3.9), labeling experiments were performed on the manually tracked data (with ground truth labeling). For a test sequence, frames from all the other seven sequences were used to learn the model parameters (mean and covariance of Gaussian). Figure 3.11 (a) shows the statistics of the number of body parts present. Figure 3.11 (b) shows the correct labeling rate vs. the number of body parts present, with the overall correct labeling rate 85.89%. From Figure 3.11 (b), we see that the correct labeling rate goes up as the number of detected body parts increases, which is consistent with the fact that with more body parts present, the probability decomposition is a more accurate approximation.

3.6.3 Detection and localization

The two detection strategies described in section 3.2 were run on the testing set. Figure 3.12 (a) shows the receiver operating characteristics (ROC) curves when the type (3) sequences were used as positive examples and type (2) sequences were used as negative examples. Figure 3.12 (b) shows ROC curves from type (1) and type

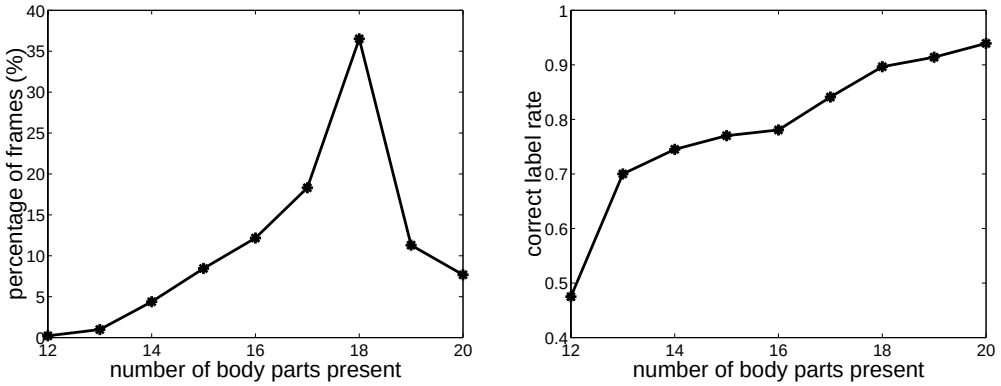


Figure 3.11: **(a)** percentage of frames corresponding to the number of body parts present in the hand-constructed data set; **(b)** correct labeling rate vs. the number of body parts present. The chance level of a body part being assigned a correct candidate feature is around 0.06. The correct rates here are much higher than that.

(2) sequences. The solid lines are results of using the sum-over-all-labelings detection strategy, and the dashed lines are of the winner-take-all strategy. This figure shows that the sum-over-all-labelings strategy performs better than the winner-take-all strategy for the gray-scale images, which is opposite to the results in section 3.5. We postulate that this is because, for gray-scale images, there are many close candidate features for one body part (Figure 3.10) and therefore there are many labelings close to the 'correct' labeling, which makes the sum-over-all-labelings strategy a closer approximation.

Figure 3.10 gives the localization results. For each image, the white dots give the best labeling. For most frames, the person is localized correctly. However, for some frames, the features consisting of the best configuration can be far away from each other, e.g., the third image in the top row (Figure 3.10). A detailed study finds that the program took the two dots on the wall as 'left elbow and left wrist', and the four dots on the chair as 'left outside knee, left ankle, left toe and left heel'. This is because for the triangulated decomposition in Figure 3.9, if 'left shoulder and left hip' are missing, then both 'left elbow and left wrist' and 'left outside knee, left ankle, left toe and left heel' are disconnected with other body parts. Therefore, the optimal labeling is composed of several independent components, possibly far away from each

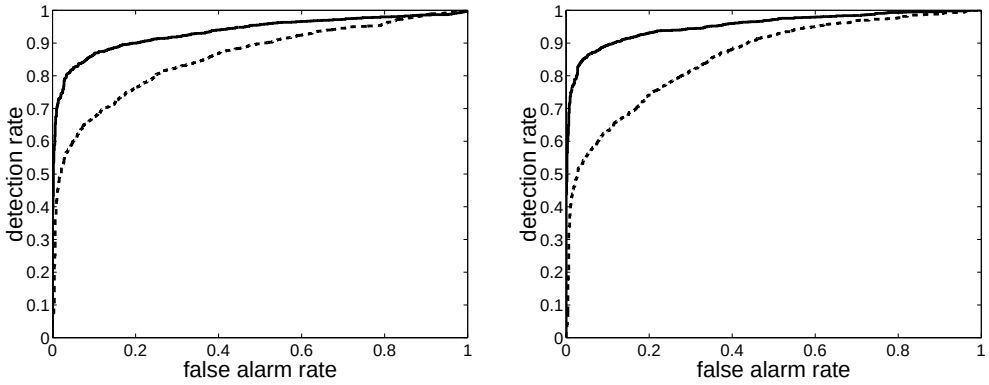


Figure 3.12: ROC curves. (a) Results of images with body and chair vs. images with chair only. (b) Results of images with body only vs. images with chair only. Solid line: the sum-over-all-labelings detection strategy; dashed line: the winner-take-all detection strategy.

other. It is clear that in this case the conditional independence required by equation (3.7) is not a good approximation any longer. We will address more on this problem later in sections 5.4.2 and 7.5.

3.6.4 Using information from multiple frames

Here we tested how the detection rates improved by integrating information over time, using the approach described in section 3.3. Type (3) and type (1) sequences were used. Figure 3.13(a) shows ROC curves of using 1 to 4 pairs of frames, respectively. Figure 3.13(b) plots the detection rates (with $P_{detect} = 1 - P_{false-alarm}$) vs. the number of frames integrated. With more frames used, the detection rate gets higher. The detection rate is more than 98% when more than 7 frames (around 200 ms) were used.

3.7 Summary

In this chapter, the detection and labeling algorithms are extended to deal with occlusion and clutter. In case of occlusion, a way of estimating approximately the foreground probability density function is presented, which allows one to find the best

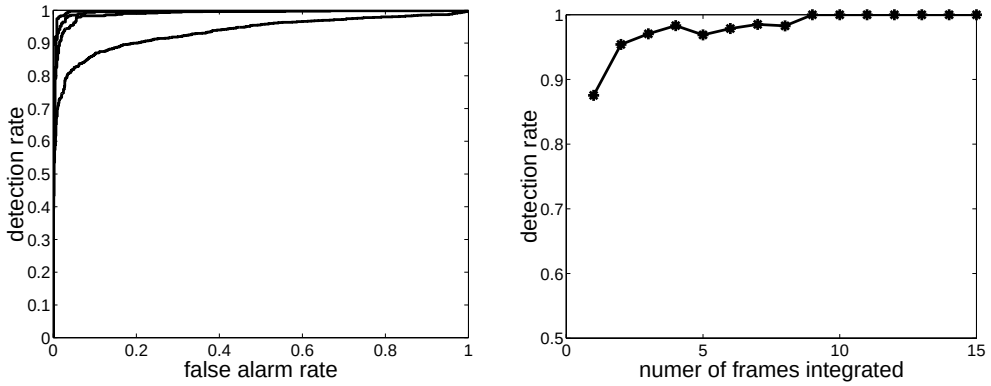


Figure 3.13: Results of integrating multiple frames. **(a)** Four curves are ROCs of integrating 1 to 4 pairs of frames, respectively. The more frames integrated, the better the ROC curve is. **(b)** detection rate (when $P_{detect} = 1 - P_{false-alarm}$) vs. number of frames used.

labeling efficiently. We also present two detection strategies: winner-take-all and sum-over-all-labelings. The algorithms have been tested and compared on motion capture data and gray-scale images. For our data sets, the winner-take-all strategy works better for motion capture data, and the sum-over-all-labelings strategy works better for gray-scale image sequences.

Chapter 4 Search of optimal decomposable triangulated graph

In the previous chapters, the graph structure is hand-crafted by expert experience (or intuition). This is not completely satisfactory 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. Therefore algorithms which can find the optimal structure automatically from data are desired. Unfortunately, the problem of finding the optimal decomposable triangulated graph is NP hard (see chapter 7 for the justification of this statement). However, we can find approximate solutions to the optimal. Two ways to build a decomposable triangulated graph automatically from labeled training data, with known correspondence between the parts and the observed features (e.g., data from a motion capture system), are presented in this chapter. One way is to grow the graph greedily according to the optimization criterion presented in section 4.1. Another way is to obtain the decomposable triangulated graph from the maximum spanning tree by adding edges, which also proves that decomposable triangulated graphs are more powerful than trees. The algorithms on labeled data lay the foundation for dealing with unlabeled training data in Chapter 5.

4.1 Optimization criterion

Our goal is to find the decomposable triangulated graph which can best describe the data. Before giving the optimization criterion, let us review the notations of 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. As we have described in section 2.2, 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

$$\begin{aligned} & P_{whole}(X_{S1}, X_{S2}, \ldots X_{SM}) \\ = & \prod_{t=1}^{T-1} P_{A_t|B_tC_t}(X_{A_t}|X_{B_t}, X_{C_t}) \cdot P_{A_TB_TC_T}(X_{A_T}, X_{B_T}, X_{C_T}) \end{aligned} \tag{4.1}$$

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 one elimination order for the decomposable graph.

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 $\overline{X}^n = (X_{S1}^n, \ldots, X_{SM}^n)$, $1 \leq n \leq N^*$, and X_{Si}^n, $1 \leq i \leq M$ is the measurements of body part Si. We call such $\overline{X}^n$ **labeled data**†, since the correspondence of 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 $\mathcal{X}$. 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 $P(\mathcal{X}|G)$. By equation (4.1), $P(\mathcal{X}|G)$ can be computed as follows,

*In this and next chapters, N is the number of samples (pairs of frames) available in the training set, which is different from that in Chapters 2 and 3.

†Note $\overline{X}^n$ in this chapter is different from other chapters. 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 chapters, it denotes the observed measurements which include body parts and background clutter.

$$
\begin{aligned}
& \log P(\mathcal{X}|G) \\
= \; & \log P(\overline{X}^1, \overline{X}^2, \ldots, \overline{X}^N|G) \\
= \; & \sum_{n=1}^{N} \log P(\overline{X}^n|G) \\
= \; & \sum_{n=1}^{N} (\sum_{t=1}^{T-1} \log P(X_{A_t}^n|X_{B_t}^n, X_{C_t}^n) + \log P(X_{A_T}^n, X_{B_T}^n, X_{C_T}^n)) \qquad (4.2) \\
= \; & \sum_{t=1}^{T-1} \sum_{n=1}^{N} \log P(X_{A_t}^n|X_{B_t}^n, X_{C_t}^n) + \sum_{n=1}^{N} \log P(X_{A_T}^n, X_{B_T}^n, X_{C_T}^n)) \\
\cong \; & N \cdot \sum_{t=1}^{T-1} E(\log P(X_{A_t}|X_{B_t}, X_{C_t})) + N \cdot E(\log P(X_{A_T}, X_{B_T}, X_{C_T})) \qquad (4.3) \\
= \; & -N \cdot \sum_{t=1}^{T-1} h(X_{A_t}|X_{B_t}, X_{C_t}) - N \cdot h(X_{A_T}, X_{B_T}, X_{C_T}) \qquad (4.4) \\
= \; & -N \cdot \sum_{t=1}^{T} h(X_{A_t}|X_{B_t}, X_{C_t}) - N \cdot h(X_{B_T}, X_{C_T}) \qquad (4.5) \\
= \; & N \cdot \sum_{t=1}^{T} I(X_{A_t}; X_{B_t}, X_{C_t}) + I(X_{B_T}; X_{C_T}) - N \cdot (\sum_{t=1}^{T} h(X_{A_t}) + h(X_{B_T}) + h(X_{C_T})) \\
& \qquad (4.6)
\end{aligned}
$$

where $E(\cdot)$ is expectation, $h(\cdot)$ is differential entropy or conditional differential entropy [24] (we consider continuous random variables here), and $I(\cdot;\cdot)$ is the mutual information between variables. Equation (4.3) is an approximation which converges to equality for $N \to \infty$ due to the weak Law of Large numbers, and equations (4.4, 4.5 and 4.6) are from the definitions and properties of differential entropy and mutual information [24, 20, 25, 18, 21]. 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 different structures, then the total number of possible structure is $\frac{1}{2}M! \cdot \prod_{j=1}^{M-2}(2j-1)$, 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}.

If the set of parts $\mathcal{S}$ is fixed, then for different probability structures, the last

term of equation (4.6), $\sum_{t=1}^{T} h(X_{A_t}) + h(X_{B_T}) + h(X_{C_T})$, is a constant, since it is the summation of the differential entropies of all the body parts. Therefore the optimization can be performed either over equation (4.5) or over the first two terms of equation (4.6), the summation of mutual information. In the next section, we use equation (4.5) for computational convenience.

4.2 Greedy search

The search for the optimal decomposable triangulated graph is a NP hard problem (we will explain it in more detail in section 7.5). We develop a greedy algorithm to grow the graph. We start from a single vertex, and add vertices one by one in a greedy way according to equation (4.5). For each possible choice of C_T (the last vertex of the last triangle), find the 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 $-h(X_{A_T}|X_{B_T}, X_{C_T})$. Add edges (A_T, B_T) and (A_T, C_T) to the graph. 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 all the 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. There is, of course, no guarantee that the global optimal solution will be found. The effectiveness of the algorithm will be explored through experiments.

4.3 Construction from a maximum spanning tree

Another way to construct decomposable triangulated graphs is adding edges to trees. In this section, we first present a way of transforming a tree into a decomposable triangulated graph. Based on that, we show how to build a decomposable triangulated graph from a maximum spanning tree in an effort to maximize the likelihood.

4.3.1 Transforming trees into decomposable triangulated graphs

Let's first recall the definitions of decomposable triangulated graphs and trees. A *decomposable triangulated graph* is a collection of cliques 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); (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 left. A *tree* is a collection of cliques of size two, where there is an elimination order of vertices such that (1) when a vertex is deleted, it is only connected with one other vertex (we call it a leaf); (2) after eliminating one leaf and the edge associated with it, the remaining subgraph is again a collection of cliques of size two until only one edge left.

Comparing the above two definitions, we can get a way of transforming trees into decomposable triangulated graphs. For an elimination order of vertices of trees, when a leaf is deleted, we can connect it with one of the other neighbors of its parent so that it is contained in a triangle. By adding these edges, a tree is turned into a collection of cliques of three and the conditions for decomposable triangulated graphs are satisfied. Figure 4.1 shows an example. The elimination oder of the vertices is $A, E, F, G, C, J, H, I, D, B$. The added edges are shown in dashed lines.

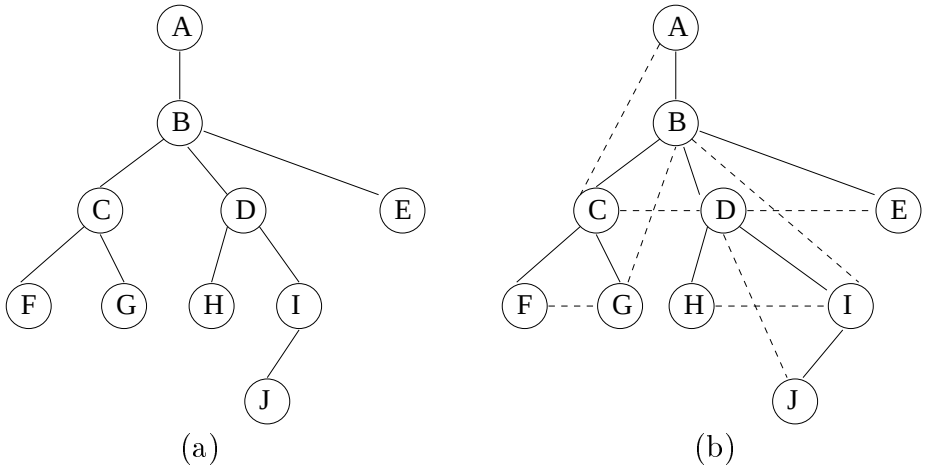


Figure 4.1: An example of transforming a tree into a decomposable triangulated graph. Figure (a) shows the tree; figure (b) gives a decomposable triangulated graph obtained by adding edges to the tree in (a).

From this procedure, the likelihood of a decomposable graph can be viewed as the summation of two parts: the likelihood associated with the tree and the likelihood gain from the tree to the triangulated graph (the likelihood gain is non-negative

because $I(X;Y,Z) \geq I(X;Y)$ for any random variables X, Y, Z). We will describe how to maximize these two parts below.

4.3.2 Maximum spanning tree

We use the same notations as in section 4.1. For given data $\mathcal{X}$, we want to find the tree G_{tree} with the highest log-likelihood $\log P(\mathcal{X}|G_{tree})$. Let $G_{tree} = (E, V)$, where E is the set of edges and V is the set of vertices (body parts). For any edge $e \in E$, let A_e and B_e denote the two vertices at the two ends. From similar derivations to those in section 4.1, we have

$$\log P(\mathcal{X}|G_{tree}) \cong N \cdot \sum_{e \in E} I(X_{A_e}; X_{B_e}) - N \cdot \sum_{v \in V} h(X_v) \tag{4.7}$$

Therefore, if we take $I(X_{A_e}; X_{B_e})$ as the value associated with each edge, the tree with the highest $\log P(\mathcal{X}|G_{tree})$ can be found by a maximum spanning tree algorithm, for example Prim's algorithm ([26]).

4.3.3 Greedy transformation

We use a greedy strategy to try to maximize the gain from a tree to a decomposable triangulated graph. Comparing equations (4.6) and (4.7), the mutual information gain by adding an edge (A_e, C) is $I(A_e; B_e, C) - I(A_e; B_e)$, where C is the vertex selected to connect with A_e when A_e is deleted (A_e must be a leaf then). There are $M - 2$ edges to be added. We will add edges in a greedy way, that is, the mutual information gain is maximized when each edge is added. Let $G_{current}$ denote the current graph at each stage. The initial value for $G_{current}$ is the maximum spanning tree obtained in the previous subsection. One leaf with its edge is deleted from $G_{current}$ at each stage. For each leaf node l of $G_{current}$, let $\pi(l)$ denote the parent of l, and $CPI(l)$ the set of nodes connected to $\pi(l)$ but excluding l. The algorithm of selecting these $M - 2$ edges can be described as follows:

At each stage t, $t = 1, \ldots, M - 2$,

For each leaf node l of $G_{current}$,

For each node $v \in CPI(l)$,

compute the gain $TG(l, v)$ of connecting l and v,

$TG(l, v) = I(l; \pi(l), v) - I(l; \pi(l))$.

Find $v^*(l) = \arg\max_v TG(l, v)$ and $g^*(l) = TG(l, v^*(l))$

Find $l^* = \arg\max_l g^*(l)$, then the selected edge is $(l^*, v^*(l^*))$ and the gain is $g^*(l^*)$.

Delete vertex l^* and its associated edge from $G_{current}$.

By adding all the selected edges to the maximum spanning tree, we construct a decomposable triangulated graph. The likelihood of this decomposable triangulated graph is the likelihood of the tree plus the summation of mutual informations gains from all the added edges. For the decomposable triangulated graph obtained in this way, we can guarantee that its likelihood is not worse than the likelihood of the optimal tree.

4.4 Computation of differential entropy - translation invariance

In the greedy search algorithm in section 4.2, we need to compute $h(X_{A_t}|X_{B_t}, X_{C_t}) = h(X_{A_t}, X_{B_t}, X_{C_t}) - h(X_{B_t}, X_{C_t})$, $1 \leq t \leq T$. For the method in section 4.3, we need the differential entropy of each single body part. If we assume that the pose and motion of the body parts are jointly Gaussian distributed, 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 [24].

In our applications, position and velocity are used as measurements for each body part, but humans can be present at different locations of the scene. In order to make the Gaussian assumption reasonable, translations need to be removed. Therefore, we use a local coordinate system ([27]) 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

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

In the greedy search algorithm, the differential entropy of all the possible triplets are needed and different triplets have different origins. We use

$$\mu' = \frac{1}{N}\sum_{n=1}^{N} \overline{\mathbf{x}}'^n = \frac{1}{N}\sum_{n=1}^{N} W\overline{\mathbf{x}}^n = W \cdot \frac{1}{N}\sum_{n=1}^{N} \overline{\mathbf{x}}^n = W\mu \tag{4.9}$$

and

$$\Sigma' = W\Sigma W^T \tag{4.10}$$

From the above equations, we can 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 (4.9) and (4.10) to get the mean and covariance for $(X_{A_t}, X_{B_t}, X_{C_t})$. 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. For a single body part, we use only velocity information to compute its differential entropy.

4.5 Experiments

We conduct experiments on labeled motion capture data. Under Gaussian assumption, we first estimated the joint probability density function (mean and covariance) of the data (sequence W3). From the estimated mean and covariance, we can compute differential entropies for all the possible triplets and pairs and further run the greedy search algorithm (section 4.2) to find the approximated best triangulated model. We also obtain a maximum spanning tree and construct a decomposable triangulated

graph from it (section 4.3). Figure 4.2 displays the models. Figure 4.2(a) is the hand-constructed model used in previous chapters (Figure 2.3(a)); (b) is the model obtained from greedy search (section 4.2); (c) is the decomposable triangulated model grown from a maximum spanning tree (section 4.3). The solid lines are edges from the maximum spanning tree and the dashed lines are added edges. (d) shows a randomly generated decomposable triangulated model, which is grown in the following way. 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 selected as its parent edge, then the newly selected vertex is connected with the two vertices of the edge.

Figure 4.3(a) shows the expected likelihood (differential entropy) of the estimated joint pdf, for each one of the models as well as a number of randomly generated models. The decomposable triangulated model from the greedy search (section 4.2) has the highest expected 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, but is superior to almost all the random triangulated models. The full Gaussian joint pdf shown for comparison has the highest likelihood. We conclude that, as far as model likelihood is concerned, there is a significant advantage for models generated by greedy search, rather than by other methods, or at random.

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 decomposable triangulated independence. To accomplish this we generate a random decomposable triangulated model, then generate data according to this model. 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 data. Figure 4.4(a) shows the expected likelihood using 50 synthetic datasets, 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 bars are the expected likelihoods of random triangulated models. To see the results

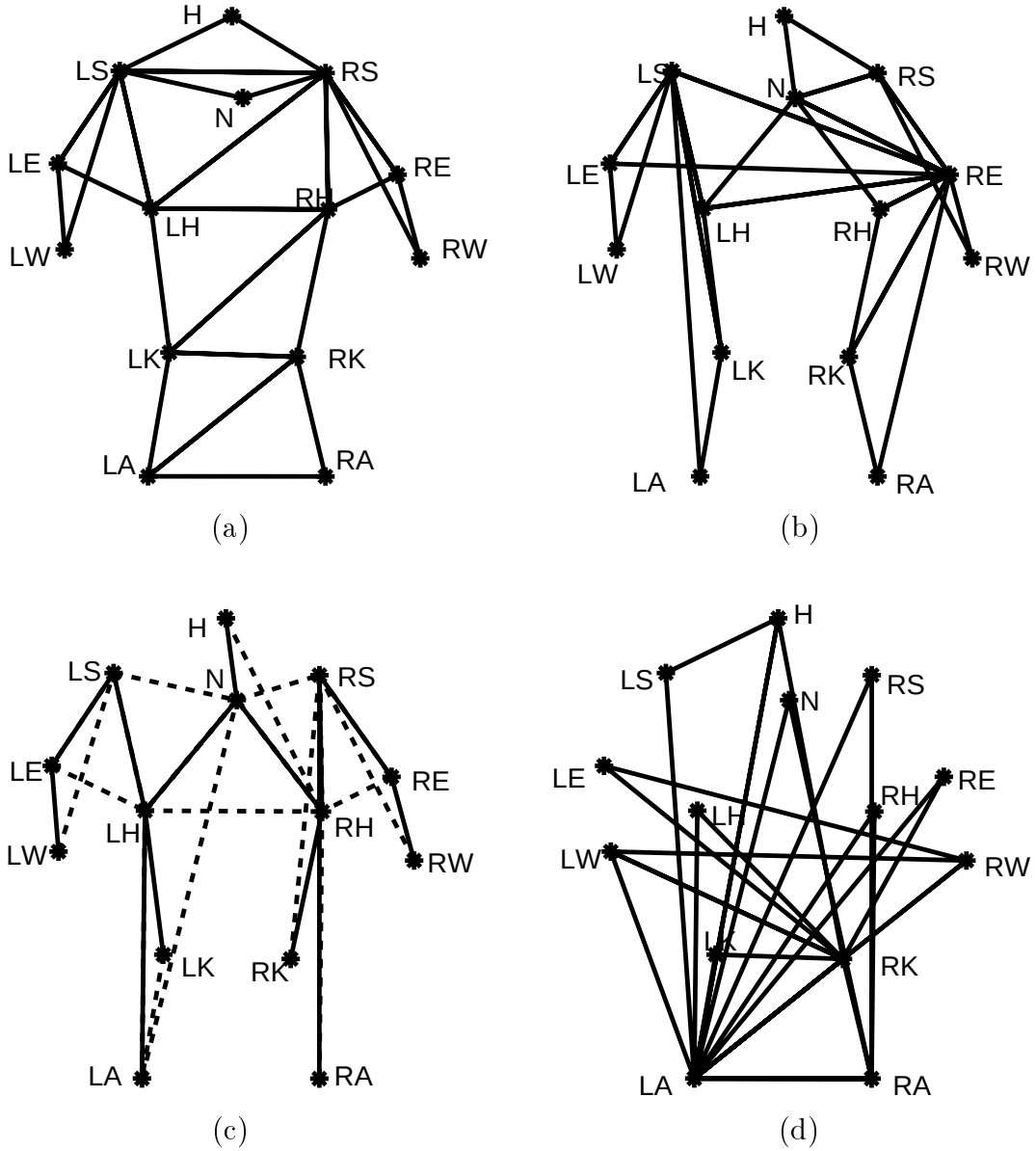


Figure 4.2: Decomposable triangulated models for motion capture data. **(a)** hand-constructed model; **(b)** model obtained from greedy search (section 4.2); **(c)** decomposable triangulated model grown from a maximum spanning tree (section 4.3). The solid lines are edges from the maximum spanning tree and the dashed lines are added edges. **(d)** a randomly generated decomposable triangulated model.

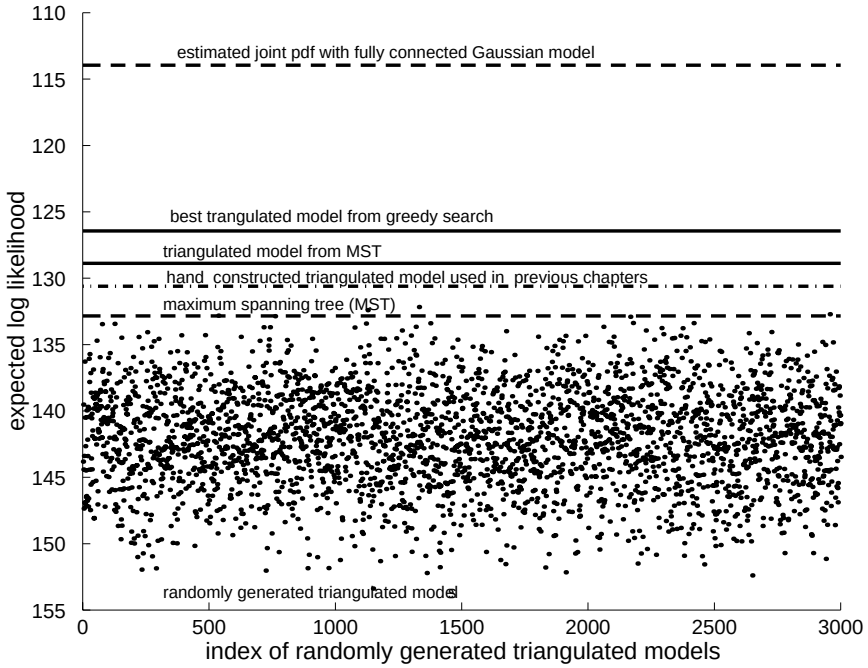
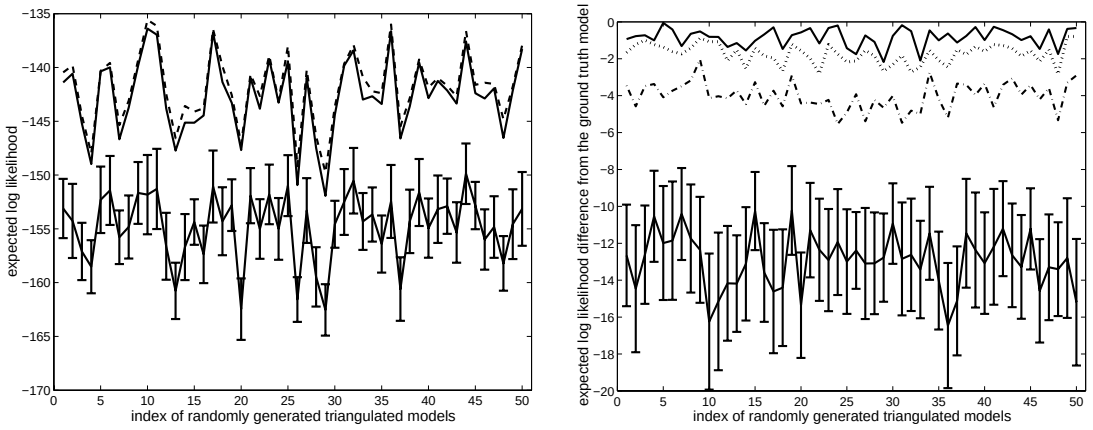


Figure 4.3: Likelihood evaluation of graph growing algorithms.



(a) (b)

Figure 4.4: Evaluation of the algorithms on synthetic data with decomposable triangulated independence. **(a)** Expected likelihoods of the true models (dashed curve) and of models from greedy search (solid curve). The solid line with error bars are the expected likelihoods of random triangulated models. **(b)** Expected likelihood difference from the respective true model, i.e., the results of subtracting the likelihood of the true model. Solid: models from the greedy search (section 4.2); dotted: triangulated models from MST (section 4.3); dash-dot: MST. The solid line with error bars are the results of random triangulated models.

more easily, Figure 4.4(b) shows the expected likelihood difference from the respective true model, i.e., the results of subtracting the likelihood of the true model. Similar to the results shown in Figure 4.3, for all the synthetic data used here, the models from the greedy search (section 4.2) have the highest likelihood (solid curve in Figure 4.4(b)), the triangulated models from maximum spanning trees (dotted curve) come next, and both are better than the maximum spanning tree models (dash-dot curve). The solid line with error bars are the results of random decomposable triangulated models. We conclude that the greedy search algorithm (section 4.2) delivers quasi-optimal solutions on this type of data. We will therefore use this algorithm in future experiments.

4.6 Summary

This chapter addresses the learning problem when the training features are labeled. Two ways of building suboptimal decomposable triangulated graphs from data have been presented and tested. It has also been shown that by the likelihood criterion the decomposable triangulated graphs obtained have a significant advantage over the optimal tree. We conclude that the greedy search algorithm (section 4.2) performs better than other methods. Hence it will be used in future experiments.

Chapter 5 Unsupervised learning of the graph structure

In Chapter 4, the training data are *labeled* in the sense that the parts of the model and the correspondence between the parts and the observed features are known. However, when we run a feature detector (such as the Lucas-Tomasi-Kanade detector [1]) on real-image sequences, the detected features are *unlabeled*, meaning that they can be from target objects and background clutter with no identity attached to each feature, and the correspondence between the candidate features and the parts of the object is unknown. In this section, we present an algorithm to learn the probabilistic independence structure of human motion automatically from this type of unlabeled training data. 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.

Our approach is based on maximizing the likelihood of the data. Taking the labeling (part assignments) as hidden variables, a variant of the EM algorithm can be applied. In the following sections, we first derive the algorithm assuming all the foreground parts are observed for each training sample, and then generalize the algorithm to handle the case when some body parts are missing (occlusion).

5.1 Brief review of the EM algorithm

The expectation-maximization (or EM, [28, 29]) algorithm is a technique of estimating probability density functions under missing (unobserved) data. There are three types of variables in EM: observed data (denoted by $\mathbf{d}$), unobserved (hidden) variables ($\mathbf{y}$), and parameters of the probability density functions to estimate ($\boldsymbol{\theta}$). The goal is to

find $\boldsymbol{\theta}$ which can maximize,

$$\begin{aligned} L(\mathbf{d}, \boldsymbol{\theta}) &= \log[p(\mathbf{d}, \boldsymbol{\theta})] \\ &= \log[p(\mathbf{d}|\boldsymbol{\theta})] + \log[p(\boldsymbol{\theta})] \\ &= \log[\int d\mathbf{y}\ p(\mathbf{d}, \mathbf{y}|\boldsymbol{\theta})] + \log[p(\boldsymbol{\theta})] \end{aligned} \tag{5.1}$$

One possible way to maximize the above function is to take derivatives with respect to $\boldsymbol{\theta}$ and equate them to zero to obtain the optimum $\boldsymbol{\theta}$. However, due to the integration over $\mathbf{y}$, this operation is difficult in most cases. The EM algorithm provides an easier way. The main idea of EM is that it is much easier to optimize $\log[p(\mathbf{d}, \mathbf{y}, \boldsymbol{\theta})]$ if we had known the values for $\mathbf{y}$. Therefore, at **E-step** we pretend that we know the parameters $\boldsymbol{\theta}$ and get the estimation of $\mathbf{y}$; at **M-step** we pretend that we know $\mathbf{y}$ (the result of E-step can be used), and obtain the best estimate of $\boldsymbol{\theta}$. These two steps are iterated until the algorithm converges. More formally, instead of optimizing equation (5.1), we will optimize,

$$Q(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}) = \mathbf{E}[\log[p(\mathbf{d}, \mathbf{y}, \boldsymbol{\theta}_t)]\ |\mathbf{d}, \boldsymbol{\theta}_{t-1}], \tag{5.2}$$

where $\boldsymbol{\theta}_t$ is the parameter to estimate at iteration t, and $\boldsymbol{\theta}_{t-1}$ is the parameter obtained from iteration $t-1$. $Q(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1})$ is the expectation of log likelihood given the observed data and parameter values from the previous iteration. Then,

E-step: Calculate $Q(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1})$, given the parameter estimates $\boldsymbol{\theta}_{t-1}$ from the previous iteration;

M-step: Get the $\boldsymbol{\theta}_t$ which can maximize $Q(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1})$.

The above two steps are iterated until convergence. It can be proved that the procedure increases the likelihood ($L(\mathbf{d}, \boldsymbol{\theta})$) at each iteration, and it converges when a local maximum is reached. More rigorous mathematical treatment of the EM algorithm can be found at [28, 29].

5.2 Learning with all foreground parts observed

In this section we will develop an algorithm to find the best decomposable triangulated model from unlabeled data using the idea of EM. Assume that we have a data set of N samples $\mathcal{X} = \{\overline{X}^1, \overline{X}^2, \ldots, \overline{X}^N\}$. Each sample $\overline{X}^n$, $1 \leq n \leq N$, is a group of detected features at time n containing the target object. But $\overline{X}^n$ is unlabeled, meaning that the correspondence between the candidate features and the parts of the object and background clutter is unknown. We want to select the useful composite parts of the object and learn the probability independence structure of parts from $\mathcal{X}$.

For the convenience of derivation, 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, the EM algorithm can be used to learn the probability structure and parameters. Our method was inspired by [22], but 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 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

$$\begin{aligned} L &= \log P(\mathcal{X}, G) \\ &= \log P(\mathcal{X}|G) + \log P(G) \\ &= \sum_{n=1}^{N} \log P(\overline{X}^n|G) + \log P(G) \\ &= \sum_{n=1}^{N} \log \sum_{h_i^n \in H^n} P(\overline{X}^n, h^n = h_i^n|G) + \log P(G) \end{aligned} \tag{5.3}$$

where h_i^n is the ith possible labeling for $\overline{X}^n$, and H^n is the set of all such labelings. Since h_i^n is a discrete variable, summation is performed in equation (5.3) instead

of integration in equation (5.1). Optimization directly over equation (5.3) is hard, and the EM algorithm solves the optimization problem iteratively. In EM, for each iteration t, we will optimize the function,

$$\begin{aligned} Q(G_t|G_{t-1}) &= E[\log P(\mathcal{X},\mathcal{H},G_t)|\mathcal{X},G_{t-1}] \\ &= \sum_{n=1}^{N} E[\log P(\overline{X}^n,h^n,G_t)|\overline{X}^n,G_{t-1}] \\ &= \sum_{n=1}^{N}\sum_{h_i^n\in H^n} P(h^n=h_i^n|\overline{X}^n,G_{t-1})\cdot\log P(\overline{X}^n,h^n=h_i^n,G_t) \\ &= \sum_{n=1}^{N}\sum_{h_i^n\in H^n} R_i^n \log P(\overline{X}^n,h^n=h_i^n,G_t) \end{aligned} \tag{5.4}$$

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,

$$R_i^n = P(h_i^n|\overline{X}^n,G_{t-1}) = P(\overline{X}^n,h_i^n,G_{t-1}) / \sum_{h_k^n\in H^n} P(\overline{X}^n,h_k^n,G_{t-1}) \tag{5.5}$$

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 3.1 to compute $P(h_i^n, \overline{X}^n, G)$ (G is G_t in equation (5.4) and G_{t-1} in equation (5.5)). 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

$$\begin{aligned} P(\overline{X}^n,h_i^n,G) &= P(\overline{X}^n|h_i^n,G)P(h_i^n,G) \\ &= P(\overline{X}_{fg}^n|h_i^n,G)P(\overline{X}_{bg}^n|h_i^n,G)P(h_i^n|G)P(G) \end{aligned} \tag{5.6}$$

Substituting equation (5.6) into equation (5.4), we get

$$\begin{aligned}
& \sum_{n=1}^{N} \sum_{h_i^n \in H^n} R_i^n \log P(\overline{X}^n, h^n = h_i^n, G_t) \\
= & \sum_{n=1}^{N} \sum_{h_i^n \in H^n} R_i^n [\log P(\overline{X}_{fg}^n | h_i^n, G_t) + \log P(\overline{X}_{bg}^n | h_i^n, G_t) + \log P(h_i^n | G_t) + \log P(G_t)] \\
= & \sum_n \sum_{h_i^n} R_i^n \log P(\overline{X}_{fg}^n | h_i^n, G_t) + \sum_n \sum_{h_i^n} R_i^n \log P(\overline{X}_{bg}^n | h_i^n, G_t) + \\
& \sum_n \sum_{h_i^n} R_i^n \log P(h_i^n | G_t) + \sum_n \sum_{h_i^n} R_i^n \log P(G_t) \qquad (5.7)
\end{aligned}$$

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 graph structures, the last two terms of equation (5.7) do not depend on G_t. If we assume uniform background densities as in Chapter 3 and [22], 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, 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 equation (2.8). Therefore the maximization of equation (5.4) is equivalent to maximizing,

$$\begin{aligned}
Q(G_t|G_{t-1}) & \sim \sum_{n=1}^{N} \sum_{h_i^n} R_i^n \log[P(\overline{X}_{fg}^n | h_i^n, G_t)] \qquad (5.8) \\
& = \sum_{n=1}^{N} \sum_{h_i^n} R_i^n [\sum_{t=1}^{T} \log P(X_{A_t}^{ni} | X_{B_t}^{ni}, X_{C_t}^{ni}) + \log P(X_{B_T}^{ni}, X_{C_T}^{ni})] \qquad (5.9)
\end{aligned}$$

$X_{A_t}^{ni}$ is the measurements of body part A_t under labeling h_i^n for $\overline{X}^n$, and so on. For most problems, the number of possible labelings is very large (on the order of M^{n_k}), and it is computationally prohibitive to sum over all the possible h_i^n as in equation (5.9). However, 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^n's, then R_i^{n*}

can be taken as 1 and other R_i^n's as 0. Hence equation (5.9) can be approximated as

$$Q(G_t|G_{t-1}) \sim \sum_{n=1}^{N}[\sum_{t=1}^{T}\log P(X_{A_t}^{ni*}|X_{B_t}^{ni*}, X_{C_t}^{ni*}) + \log P(X_{B_T}^{ni*}, X_{C_T}^{ni*})] \quad (5.10)$$

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 3.1 using model G_{t-1}. Comparing equation (5.10) with equation (4.2) 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 greedy algorithm described in section 4.2. 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 vs. mixture of Gaussian for clustering problems ([30]). 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-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-step: the mean μ_t and covariance matrix Σ_t can be updated as

$$\mu_t = \frac{1}{N}\sum_n \overline{X}_{fg}^{n*} \quad (5.11)$$

$$\Sigma_t = \frac{1}{N}\sum_n (\overline{X}_{fg}^{n*} - \mu_t)(\overline{X}_{fg}^{n*} - \mu_t)^T \quad (5.12)$$

Use μ_t and Σ_t to compute differential entropies (section 4.4) and run the graph growing algorithm described in section 4.2 to get G_t.

Comparing with the standard EM technique, we make two approximations in the above procedure. In the E-step, we use the best labeling instead of the weighted sum of all the possible labelings. In the M-step, there is no guarantee that the graph growing algorithm will find the optimal graph. We evaluate these approximations with experiments.

5.3 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 [22], and the EM 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 EM 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

$$\mu_t = \frac{1}{N} \sum_n E(\overline{X}_{fg}^n) \tag{5.13}$$

$$\Sigma_t = \frac{1}{N} \sum_n E(\overline{X}_{fg}^n - \mu_t)(\overline{X}_{fg}^n - \mu_t)^T = \frac{1}{N} \sum_n E(\overline{X}_{fg}^n \overline{X}_{fg}^{nT}) - \mu_t \mu_t^T \tag{5.14}$$

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}.

5.4 Experiments

We tested our algorithm on both motion capture data (Johansson displays) and on features detected from real-image sequences. The motion capture data allowed us to run the learning algorithm under conditions where all body parts were present (section 5.2) 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 [1] was used to generate the training set, and with many occlusions occurring (section 5.3).

5.4.1 Results on motion capture data

We first investigate the performance of the algorithm on motion capture data. Sequence W3 was used for learning and W4 for testing (see section 2.4 for detailed description of the data). 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 (section 4.4 and [24]). We ran the EM-like algorithm described in this chapter ten times with different random initializations.

Evaluation of the EM-like algorithm. The EM algorithm guarantees that the likelihood improves with each iteration and converges. In our algorithm (section 5.2), we make two approximations: that the best hypothesis labeling is taken instead of summing over all the possible hypotheses (equation (5.10)) and a greedy search is used to find the approximated optimal graph structure. These approximations are evaluated by checking how the log-likelihoods evolve with EM iterations and if they converge. Figure 5.1 shows how the likelihood evolves with iterations. We used random initializations, and each curve of Figure 5.1 corresponds to one such random initialization. From Figure 5.1 we can see that generally the log-likelihoods grow and

converge well with the iterations of EM.

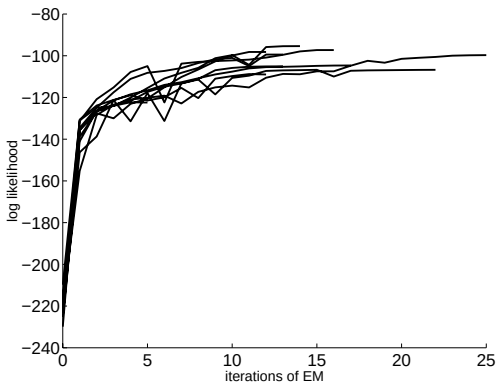


Figure 5.1: Log-likelihood vs. iterations of EM for different random initializations. Iteration 0 means random initializations, iteration 1 is after one iteration, and so on. The results are from motion capture data, assuming that all the foreground parts are observed in the learning algorithm (section 5.2).

Models obtained. Figures 5.2 (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 5.2), we can quantify the detection and labeling performance in testing.

Figure 5.3 depicts how the model in Figure 5.2(a) evolves with iterations by showing the mean positions of the parts of the model at each stage.

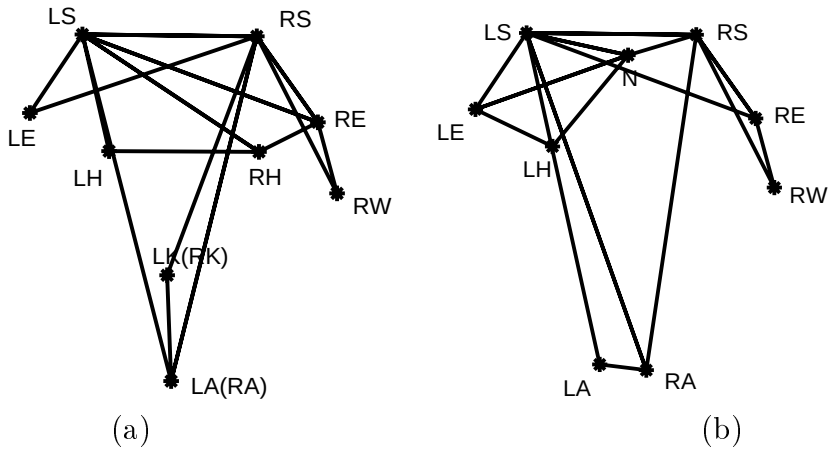


Figure 5.2: Two decomposable triangulated models for Johansson displays. These models were learned automatically from unlabeled training data. 'L': left; 'R': right. H:head, N:neck, S:shoulder, E:elbow, W:wrist, H:hip, K:knee, A:ankle.

Detection and labeling. Figure 5.4 shows the detection and labeling results by using the two models in Figure 5.2. Figures 5.4 (a) and (b) are ROC curves corresponding to Figures 5.2 (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 5.4(c) shows the detection rate vs. number of body parts displayed with regard to the fixed threshold. Figure 5.4 (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 5.4 (c) and (d), the solid lines (with *) are from model Figure 5.2 (a); the dashed lines (with o) are from model Figure 5.2 (b); and dash-dot lines with triangles are from the hand-crafted model in Figure 2.3 (a) (also see Figure 3.3). Though it is not fair to compare directly the results from the automatically learned models and the hand-constructed model due to the fact that they have different number of parts and therefore different

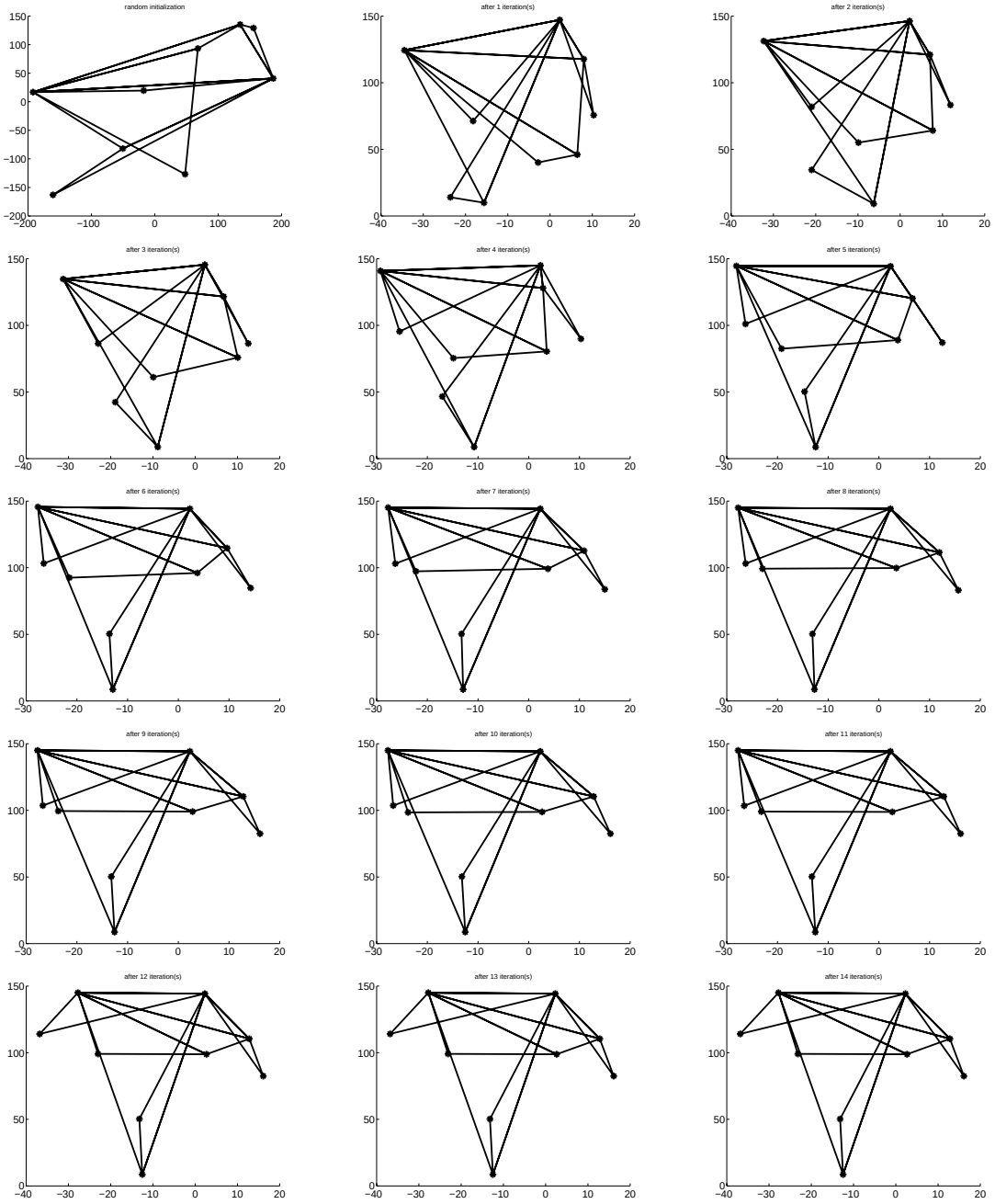


Figure 5.3: Evolution of a model with iterations (from motion capture data).

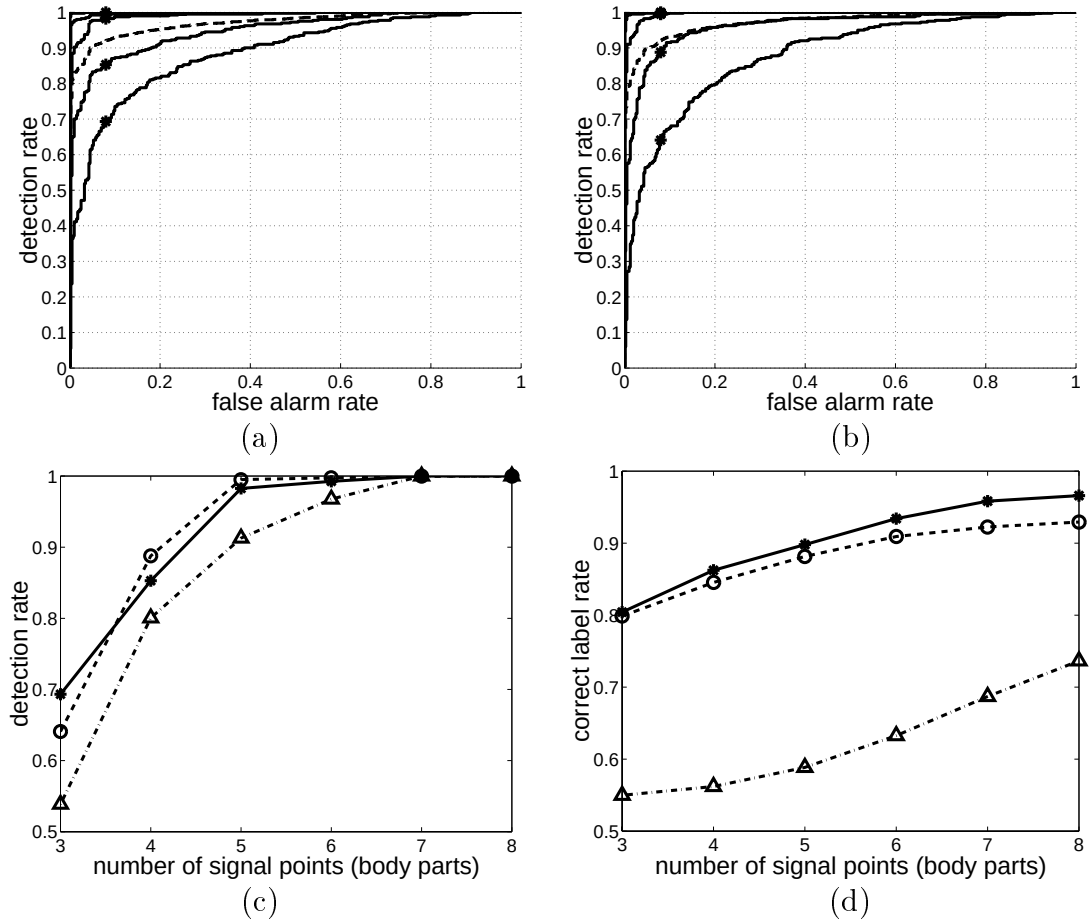


Figure 5.4: Detection and labeling results. (a) and (b) are ROC curves corresponding to models Figure 5.2 (a) and (b), respectively. Solid lines: 3 to 8 body parts with 30 background points vs. 30 background points only. The more body parts present, the better the ROC. Dashed line: overall ROC considering 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 5.2 (a); dashed lines (with o) are from model Figure 5.2 (b); and dash-dot lines with triangles are from the hand-crafted model in Figure 2.3(a) (also see Figure 3.3).

properties of graph connectivity for the same number of body parts present, Figure 5.4 still shows that the automatically learned models work quite well.

5.4.2 Results on real-image sequences

We did experiments on the same image sequences as in section 3.6, and compared the automatically learned model with the hand-constructed one.

We learned an 11-part model by taking the training data as unlabeled. Figure 5.5 shows the best model obtained (by the likelihood criterion) after we ran the EM algorithms for 12 times. Figure 5.5(a) gives the mean positions and mean velocities (shown in arrows) of the composed parts selected by the algorithm. Figure 5.5(b) shows the learned decomposable triangulated probabilistic structure. The numbers in brackets show the correspondence of (a) and (b) and one elimination order.

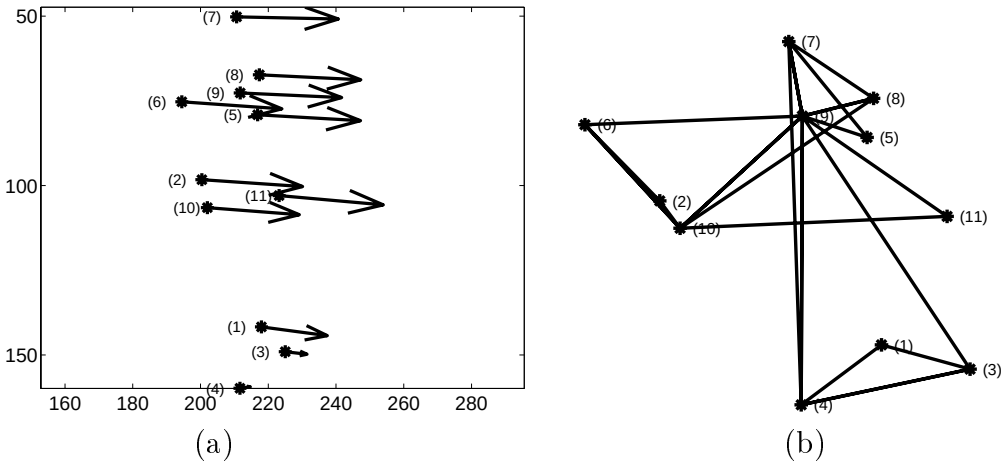


Figure 5.5: (a) The mean positions and mean velocities (shown in arrows) of the composed parts selected by the algorithm. (b) The learned decomposable triangulated probabilistic structure. The numbers in brackets show the correspondence of (a) and (b) and one elimination order.

Figure 5.6 shows labeling results on some sample frames. Comparing this figure with figure 3.10, we can see that here all the features composed of the best-configuration are on the human body, but this is not true for Figure 3.10. For experiments displayed in Figure 3.10, there exists a problem that due to occlusion the best

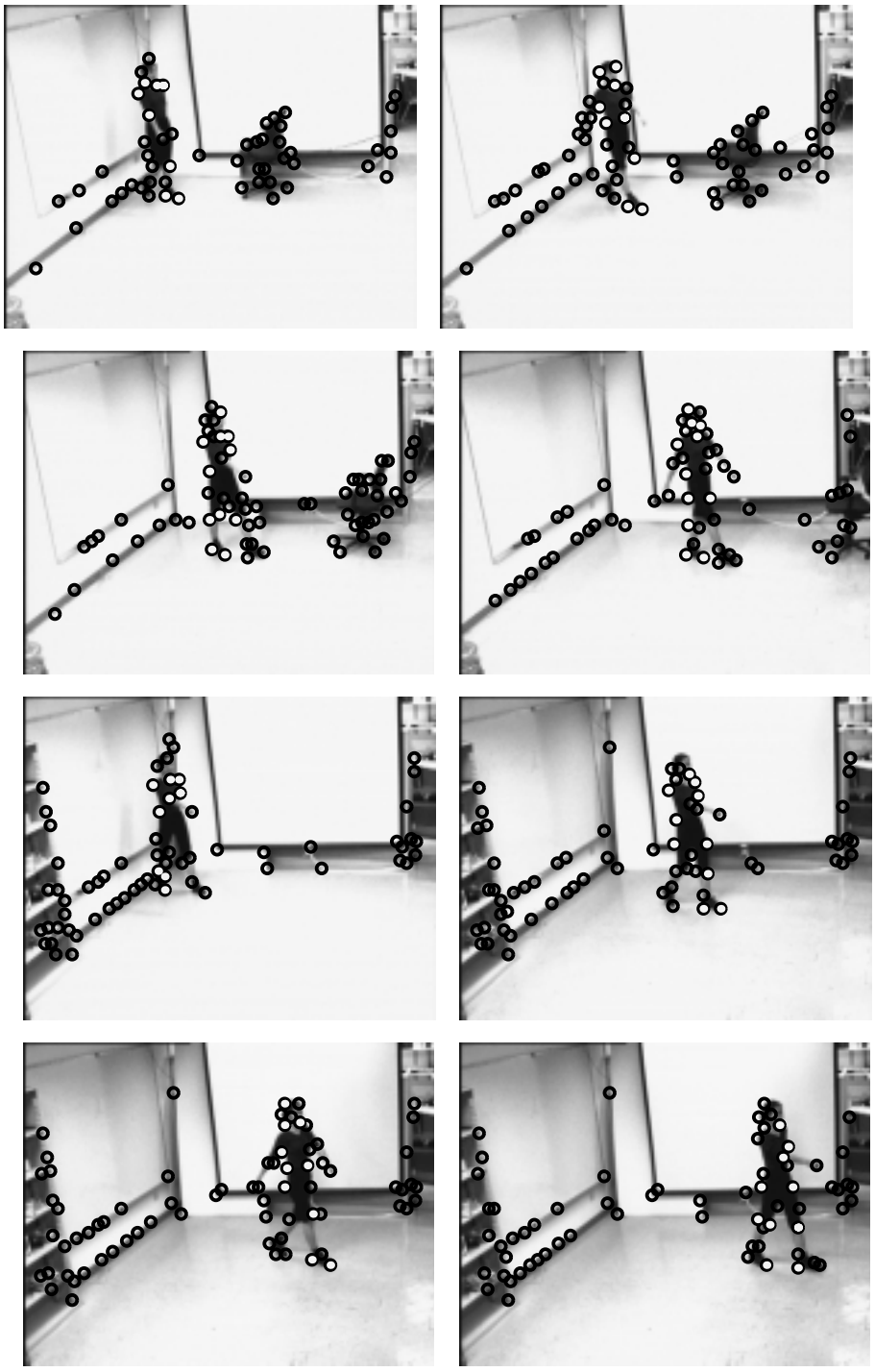
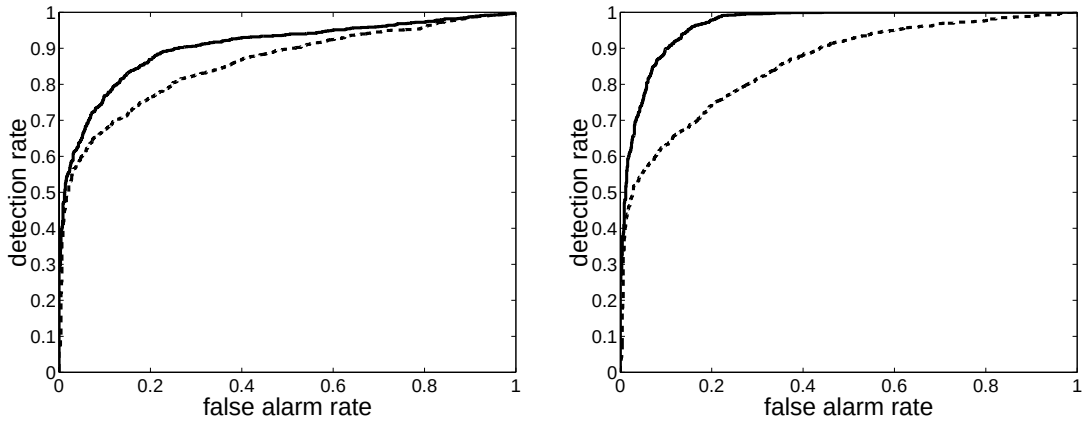


Figure 5.6: Sample frames from body and chair moving sequences (top two rows) and body moving sequences (bottom two rows). The dots (either in black or in white) are the features selected by Lucas-Tomasi-Kanade algorithm on two frames. The white dots are the most human-like configuration found by the automatically learned model (Figure 5.5).

configuration is composed of several independent components and these component can be far away from each other. The situation has been improved a lot by using the automatically learned model as in Figure 5.5. Comparing the two models (Figures 3.9 and 5.5), we can find that Figure 3.9 is a more local model, which means that the parts close to each other are connected, and Figure 5.5 is more global since some triangles contains parts far away, for example the triangle of parts (3), (4) and (7). This global connectivity helps prevent the graph from becoming separated in case of some body parts missing.



(a) (b)

Figure 5.7: 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 5.5; dashed line: using the model in Figure 3.9 (dashed lines of Figure 3.12).

The ROC curves in Figure 5.7 show the detection results. Detection is based on thresholding the likelihood of the most human-like configuration selected by the model (winner-take-all). Solid lines are from the automatically learned model as in Figure 5.5; dashed lines are from the model in Figure 3.9 (dashed lines of Figure 3.12). Figure 5.7(a) shows results of images with body and chair vs. images with chair only; and curves in Figure 5.7 (b) are results of images with body only vs. images with chair only. From Figure 5.7, we see that the automatically learned model performs better than the hand-constructed model in Figure 3.9. 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 Figure 3.9).

5.5 Summary

In this chapter, we develop an algorithm for learning the probability independence structure of parts from unlabeled data, i.e., data with unknown correspondence between the parts and the observed features, and with clutter and occlusion. A variant of the EM algorithm is developed where the labeling of the data (part assignments) is treated as hidden variables. We use decomposable triangulated graphs to depict the probabilistic independence of parts, but the unsupervised technique is not limited to this type of graph. Our algorithm enables the creation of systems that are able to learn models of human motion completely automatically from real-image sequences.

Chapter 6 Mixtures of decomposable triangulated models

In the previous chapters, we model each triangle by a Gaussian distribution, therefore the joint probability density function of all the parts is a unimodal 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 Gaussians [30], with each component model being a Gaussian with decomposable triangulated independence.

6.1 Definition

A mixture model is a weighted sum of several individual decomposable triangulated models. Each component model is relatively independent in the sense that different components can have different sets of body parts. More formally, a C-cluster (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,

$$P(\overline{X}) = \sum_{j=1}^{C} P(\overline{X}|c=j)P(c=j) \quad (6.1)$$

$$= \sum_{j=1}^{C} \sum_{h_{ji} \in H_j} P(\overline{X}, h_j = h_{ji}|c=j)P(c=j) \quad (6.2)$$

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 section 3.1 and equation (5.6), that is,

$$P(\overline{X}, h_j = h_{ji}|c=j) = P(\overline{X}|h_{ji}, c=j)P(h_{ji}|c=j) \quad (6.3)$$

$$= P_{G^j}(\overline{X}_{fg}|h_{ji}, c=j)P(\overline{X}_{bg}|h_{ji}, c=j)P(h_{ji}|c=j) \quad (6.4)$$

The first two terms of equation (6.4), $P_{G^j}(\overline{X}_{fg}|h_{ji}, c=j)$ and $P(\overline{X}_{bg}|h_{ji}, c=j)$ can be estimated as in section 3.1, and we assume that under one component model j, $(1 \leq j \leq C)$, the prior probabilities of possible labelings are uniformly distributed, i.e., $P(h_{ji}|c=j) = 1/|H_j|$, where $|H_j|$ is the size of H_j.

6.2 EM learning rules

For clarity, we first assume that all the foreground parts are present for each component. Compared with the EM algorithm in section 5.2, 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 its associated prior probabilities Π. By Bayes' rule and

equation (6.2), the likelihood function we want to maximize is

$$\begin{aligned} L &= \log P(\mathcal{X}, G, \Pi) \\ &= \log P(\mathcal{X}|G, \Pi) + \log P(G, \Pi) \\ &= \sum_{n=1}^{N} \log P(\overline{X}^n|G, \Pi) + \log P(G, \Pi) \\ &= \sum_{n=1}^{N} \log \sum_{j=1}^{C} \sum_{h_{ji}^n \in H_j^n} P(\overline{X}^n, h_j^n = h_{ji}^n, c^n = j|G, \Pi) + \log P(G, \Pi) \end{aligned} \tag{6.5}$$

where h_{ji}^n is the ith possible labeling of $\overline{X}^n$ under the jth component model, and H_j^n is the set of all such possible labelings. Optimization directly over equation (6.5) is hard, and the EM algorithm solves the problem iteratively. Let $G_t = [G_t^1 G_t^2 \cdots G_t^C]$ and $\Pi_t = [\pi_t^1 \pi_t^2 \cdots \pi_t^C]$ denote the parameters at iteration t. Then in EM, at each iteration t, we will optimize the function,

$$\begin{aligned} & Q(G_t, \Pi_t|G_{t-1}, \Pi_{t-1}) \\ = \; & E[\log P(\mathcal{X}, \mathcal{H}, \mathcal{C}, G_t, \Pi_t)|\mathcal{X}, G_{t-1}, \Pi_{t-1}] && (6.6) \\ = \; & \sum_{n=1}^{N} E[\log P(\overline{X}^n, h^n, c^n, G_t, \Pi_t)|\overline{X}^n, G_{t-1}, \Pi_{t-1}] && (6.7) \\ = \; & \sum_{n=1}^{N} \sum_{j=1}^{C} \sum_{h_{ji}^n \in H_j^n} P(h_j^n = h_{ji}^n, c^n = j|\overline{X}^n, G_{t-1}, \Pi_{t-1}) \\ & \qquad \cdot \log P(\overline{X}^n, h_j^n = h_{ji}^n, c^n = j, G_t, \Pi_t) && (6.8) \\ = \; & \sum_{n=1}^{N} \sum_{j=1}^{C} \sum_{h_{ji}^n \in H_j^n} P(h_j^n = h_{ji}^n|c^n = j, \overline{X}^n, G_{t-1}, \Pi_{t-1}) \cdot P(c^n = j|\overline{X}^n, G_{t-1}, \Pi_{t-1}) \\ & \qquad \cdot \log P(\overline{X}^n, h_j^n = h_{ji}^n, c^n = j, G_t, \Pi_t) && (6.9) \end{aligned}$$

The $E[\cdot]$ in equations (6.6) and (6.7) is the expectation of log likelihood given the observed data and parameters from iteration $t-1$. Equation (6.8) is computing the expectation by summing over all the possible values of the hidden variables. For convenience, we define $R_{ji}^n = P(h_j^n = h_{ji}^n|c^n = j, \overline{X}^n, G_{t-1}, \Pi_{t-1})$, which is the probability of a labeling h_{ji}^n of $\overline{X}^n$ given $\overline{X}^n$ and $\overline{X}^n$ belonging to cluster j, and $\omega_j^n = P(c^n = j|\overline{X}^n, G_{t-1}, \Pi_{t-1})$, which is the probability of $\overline{X}^n$ belonging to cluster j

given $\overline{X}^n$. We explain how to compute ω_j^n and R_{ji}^n below.

$$
\begin{aligned}
& \omega_j^n \\
= \; & P(c^n = j|\overline{X}^n, G_{t-1}, \Pi_{t-1}) \\
= \; & \frac{P(c^n = j, \overline{X}^n|G_{t-1}, \Pi_{t-1})}{\sum_{k=1}^{C} P(c^n = k, \overline{X}^n|G_{t-1}, \Pi_{t-1})} \\
= \; & \frac{P(\overline{X}^n|c^n = j, G_{t-1}, \Pi_{t-1})P(c^n = j|G_{t-1}, \Pi_{t-1})}{\sum_{k=1}^{C} P(\overline{X}^n|c^n = k, G_{t-1}, \Pi_{t-1})P(c^n = k|G_{t-1}, \Pi_{t-1})} \\
= \; & \frac{\sum_{h_{ji}^n \in H_j^n} P(\overline{X}^n, h_{ji}^n|c^n = j, G_{t-1}, \Pi_{t-1}) \cdot \pi_{t-1}^j}{\sum_{k=1}^{C} \sum_{h_{ki}^n \in H_k^n} P(\overline{X}^n, h_{ki}^n|c^n = k, G_{t-1}, \Pi_{t-1}) \cdot \pi_{t-1}^k} \\
= \; & \frac{\pi_{t-1}^j \sum_{h_{ji}^n \in H_j^n} P(\overline{X}^n|h_{ji}^n, c^n = j, G_{t-1}, \Pi_{t-1})P(h_{ji}^n|c^n = j, G_{t-1}, \Pi_{t-1})}{\sum_{k=1}^{C} \pi_{t-1}^k \sum_{h_{ki}^n \in H_k^n} P(\overline{X}^n|h_{ki}^n, c^n = k, G_{t-1}, \Pi_{t-1})P(h_{ki}^n|c^n = k, G_{t-1}, \Pi_{t-1})} \qquad (6.10) \\
= \; & \frac{\pi_{t-1}^j \sum_{h_{ji}^n \in H_j^n} P(\overline{X}_{fg(ji)}^n|h_{ji}^n, G_{t-1}^j)}{\sum_{k=1}^{C} \pi_{t-1}^k \sum_{h_{ki}^n \in H_k^n} P(\overline{X}_{fg(ki)}^n|h_{ki}^n, G_{t-1}^k)} \qquad (6.11)
\end{aligned}
$$

where $\overline{X}_{fg(ki)}^n$, $k = 1, \ldots, C$, is the foreground measurements of labeling $h_{ki}^n \in H_k^n$ under component model k. The first couple of steps in the above derivation are mainly from Bayes' rule and distributive law of summation. The equal sign from equation (6.10) to equation (6.11) holds due to the following three reasons. (1). Given $\overline{X}^n$ belonging to cluster j, the probability of $\overline{X}^n$ only depends on G_j, not other component models or priors. Therefore $P(\overline{X}^n|h_{ki}^n, c^n = k, G_{t-1}, \Pi_{t-1}) = P(\overline{X}^n|h_{ki}^n, G_{t-1}^k) = P(\overline{X}_{fg(ki)}^n|h_{ki}^n, G_{t-1}^k)P(\overline{X}_{bg(ki)}^n|h_{ki}^n, G_{t-1}^k)$. (2). In this chapter, we assume that all the component models have the same number of body parts. Then in the case of all foreground parts observed, the number of background points is the same for different labelings under different component models. Therefore under the uniform background assumption the background probabilities $\overline{X}_{bg(ki)}^n$ can be canceled out. (3). Since all the component models have the same number of body parts, the total number of possible labelings is the same for different component models. If we assume that within one component model the prior probabilities of all the possible labelings are uniformly distributed, then $P(h_{ki}^n|c^n = k, G_{t-1}, \Pi_{t-1})$ is the same for all the possible choices of k and i.

Since each G_{t-1}^k, $k = 1, \ldots, C$, is a decomposable triangulated Gaussian model, the summation $\sum_{h_{ki}^n \in H_k^n} P(\overline{X}_{fg(ki)}^n | h_{ki}^n, G_{t-1}^k)$ in equation (6.11) can be computed efficiently by dynamic programming (use 'sum' operation instead of 'max' operation, for more details see section 3.2.2 and [31]).

The computation of R_{ji}^n is the same as equation (5.5) but using component model G_{t-1}^j. ω_j^n and R_{ji}^n are computed using the parameters from iteration $t-1$, hence they are fixed constants for function Q at iteration t.

Substituting ω_j^n and R_{ji}^n into equation (6.9), we get

$$
\begin{aligned}
& Q(G_t, \Pi_t | G_{t-1}, \Pi_{t-1}) \\
= \; & \sum_{n=1}^{N} \sum_{j=1}^{C} \sum_{h_{ji}^n \in H_j^n} R_{ji}^n \cdot \omega_j^n \cdot \log P(\overline{X}^n, h_j^n = h_{ji}^n, c^n = j, G_t, \Pi_t) \\
= \; & \sum_{n=1}^{N} \sum_{j=1}^{C} \omega_j^n \sum_{h_{ji}^n \in H_j^n} R_{ji}^n \cdot \log P(\overline{X}^n, h_j^n = h_{ji}^n, c^n = j, G_t, \Pi_t) \\
= \; & \sum_{n=1}^{N} \sum_{j=1}^{C} \omega_j^n \sum_{h_{ji}^n \in H_j^n} R_{ji}^n \cdot [\log P(\overline{X}^n | h_j^n = h_{ji}^n, c^n = j, G_t, \Pi_t) \\
& \qquad + \log P(h_j^n = h_{ji}^n | c^n = j, G_t, \Pi_t) + \log P(c^n = j | G_t, \Pi_t)] \\
= \; & \sum_{n=1}^{N} \sum_{j=1}^{C} \omega_j^n \sum_{h_{ji}^n \in H_j^n} R_{ji}^n \cdot [\log P(\overline{X}_{fg(ji)}^n | h_j^n = h_{ji}^n, c^n = j, G_t, \Pi_t) \\
& \qquad + \log P(\overline{X}_{bg(ji)}^n | h_j^n = h_{ji}^n, c^n = j, G_t, \Pi_t) \\
& \qquad + \log P(h_j^n = h_{ji}^n | c^n = j, G_t, \Pi_t) + \log P(c^n = j | G_t, \Pi_t)] \\
= \; & Q_1 + Q_2 + Q_3 + Q_4 \qquad (6.12)
\end{aligned}
$$

where

$$
\begin{aligned}
Q_1 &= \sum_{n=1}^{N}\sum_{j=1}^{C}\omega_j^n \sum_{h_{ji}^n \in H_j^n} R_{ji}^n \cdot \log P(\overline{X}_{fg(ji)}^n | h_j^n = h_{ji}^n, c^n = j, G_t, \Pi_t) \\
&= \sum_{j=1}^{C}\sum_{n=1}^{N}\omega_j^n \sum_{h_{ji}^n \in H_j^n} R_{ji}^n \cdot \log P(\overline{X}_{fg(ji)}^n | h_j^n = h_{ji}^n, G_t^j) \\
&= \sum_{j=1}^{C} Q_1^j \qquad (6.13) \\
Q_2 &= \sum_{n=1}^{N}\sum_{j=1}^{C}\omega_j^n \sum_{h_{ji}^n \in H_j^n} R_{ji}^n \cdot \log P(\overline{X}_{bg(ji)}^n | h_j^n = h_{ji}^n, c^n = j, G_t, \Pi_t) \qquad (6.14) \\
Q_3 &= \sum_{n=1}^{N}\sum_{j=1}^{C}\omega_j^n \sum_{h_{ji}^n \in H_j^n} R_{ji}^n \cdot \log P(h_j^n = h_{ji}^n | c^n = j, G_t, \Pi_t) \qquad (6.15) \\
Q_4 &= \sum_{n=1}^{N}\sum_{j=1}^{C}\omega_j^n \sum_{h_{ji}^n \in H_j^n} R_{ji}^n \cdot \log P(c^n = j | G_t, \Pi_t) \\
&= \sum_{j=1}^{C}\sum_{n=1}^{N}\omega_j^n \sum_{h_{ji}^n \in H_j^n} R_{ji}^n \cdot \log(\pi_t^j) \\
&= \sum_{j=1}^{C}(\sum_{n=1}^{N}\omega_j^n)\log(\pi_t^j) \qquad (6.16)
\end{aligned}
$$

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 equation (6.13), the best G_t^j is the one which can maximize

$$
\begin{aligned}
Q_1^j &= \sum_{n=1}^{N}\omega_j^n \sum_{h_{ji}^n \in H_j^n} R_{ji}^n \cdot \log P(\overline{X}_{fg(ji)}^n | h_j^n = h_{ji}^n, G_t^j) \qquad (6.17) \\
&\approx \sum_{n=1}^{N}\omega_j^n \log P(\overline{X}_{fg(ji)}^{n*} | G_t^j) \qquad (6.18)
\end{aligned}
$$

where $\overline{X}_{fg(ji)}^{n*}$ is the foreground configuration with the highest R_{ji}^n, i.e. the best labeling of $\overline{X}^n$ under model G_{t-1}^j. The approximation from equation (6.17) to (6.18) is under the same reasoning as from equation (5.9) to (5.10). Under Gaussian assumption, the maximum likelihood parameter estimation of G_t^j can be obtained by taking derivatives of equation (6.18) with respect to mean and covariance matrix and equating to zero. Then we have the updated parameters,

$$
\mu_t^j = \frac{\sum_n \omega_j^n \overline{X}_{fg(ji)}^{n*}}{\sum_n \omega_j^n} \qquad (6.19)
$$

$$\Sigma_t^j = \frac{\sum_n \omega_j^n \overline{X}_{fg(ji)}^{n*} (\overline{X}_{fg(ji)}^{n*})^T}{\sum_n \omega_j^n} - \mu_t^j (\mu_t^j)^T \tag{6.20}$$

From μ_t^j and Σ_t^j, the decomposable triangulated structure can be obtained by running the graph growing algorithm in section 4.

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

$$\pi_t^j = \frac{\sum_n \omega_j^n}{N} \tag{6.21}$$

The whole EM 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, $G_0 = [G_0^1, \ldots, G_0^C]$, and the initial priors Π_0. At each EM iteration t, (t from 1 till convergence),

E-step: For each $\overline{X}^n$, find the best labeling $\overline{X}_{fg(ji)}^{n*}$ using component model G_{t-1}^j, $j = 1, \ldots, C$ and compute ω_j^n by equation (6.11).

M-step: Compute μ_t^j and Σ_t^j as in equations (6.19) and (6.20). Run the graph growing algorithm (section 4) on each Σ_t^j to obtain updated G_t^j, $j = 1, \ldots, C$. Update Π_t as in equation (6.21).

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 5.3 are applied.

6.3 Detection and labeling using mixture models

For an observation $\overline{X}$, we can run the detection and labelings algorithms as in Chapter 3 using 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 winner-take-all strategy or 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 $\overline{X}_{fg(j)}^*$, $j = 1, \ldots, C$.

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)

Table 6.1: Types of images used in the experiments. '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 3 or 4 sequences, with around 80 frames for each sequence. The +/- in the code-names denotes whether movement is R-L or L-R.

6.4 Experiments

In this section, we conduct experiments on gray-scale image sequences. The image sequences were acquired using a digital cam-corder at 30 Hz 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 [1] 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 [31].

The image sequences (see Figures 6.1 and 6.6 for sample images) used in the experiments are summarized in Table 6.1. The ten subjects of the (p1) sequences include 6 males and 4 females from 20 to 50 years old. We assume that the distance between the person and the camera is constant. The different sizes of the subjects are taken care of by the probabilistic model automatically.

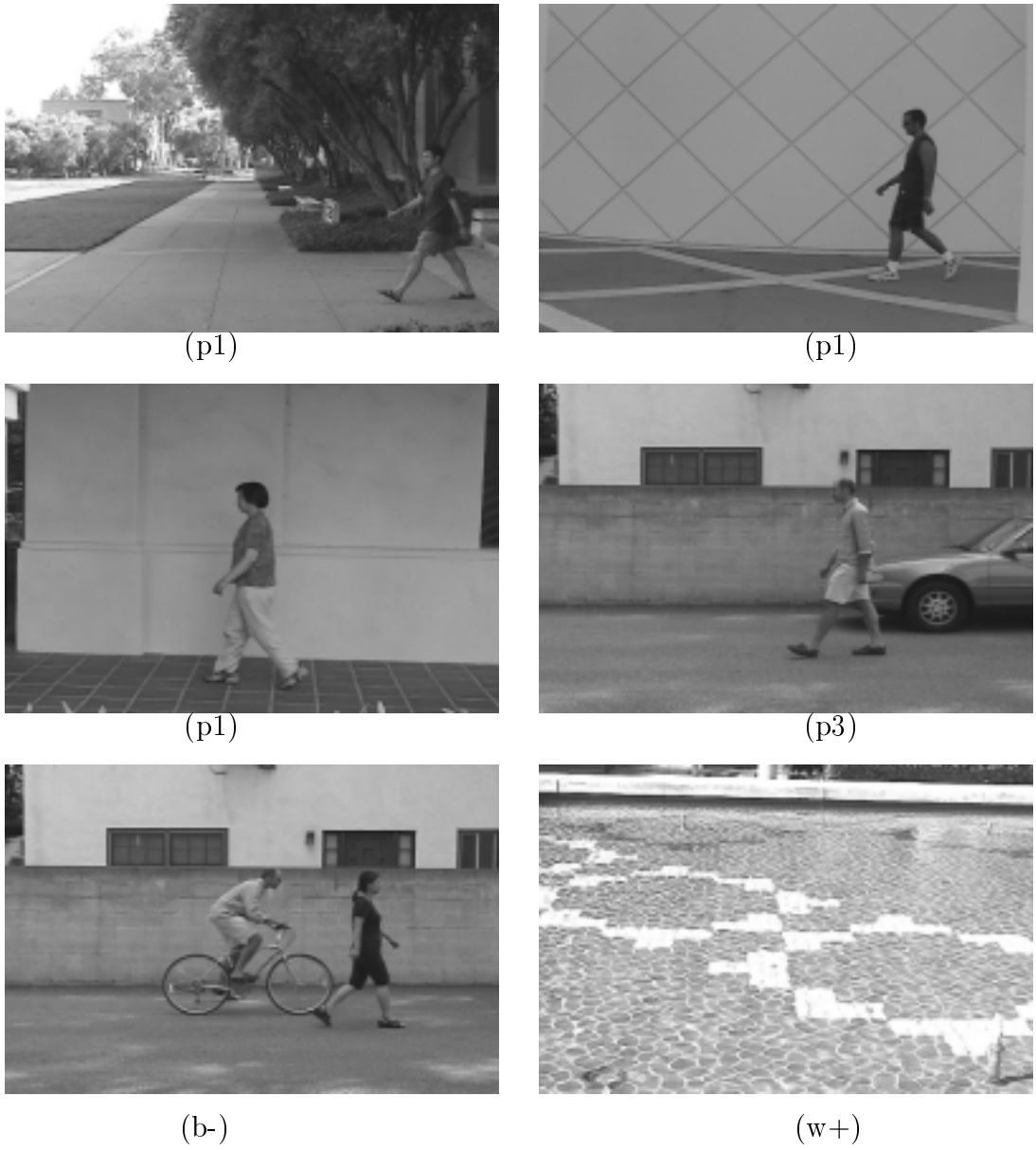


Figure 6.1: Sample images. The text string in parenthesis indicates the image type.

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.

6.4.1 Evaluation of the EM algorithm

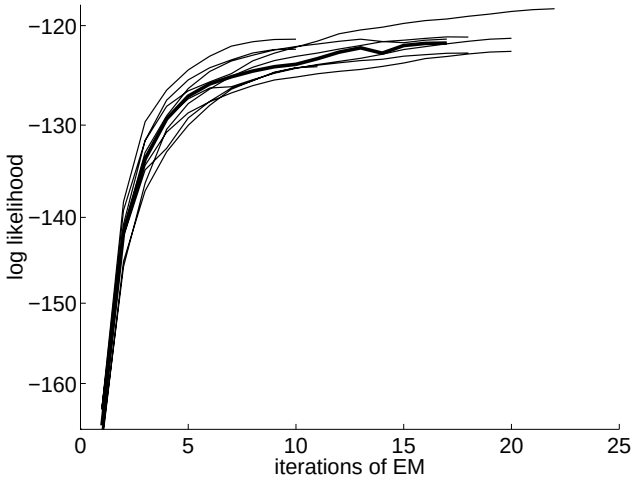
There are two approximations in the unsupervised learning algorithms (see the end of section 5.2). Here we evaluate the EM-like algorithm by checking how the log-likelihoods evolve with EM iterations and if they converge.

We learn two types of models. The first one is a single-subject model: using 9 type (p1) sequences of subject LG. The other is a multiple-people model: using 12 type (p1) sequences from 4 subjects (including subject LG).

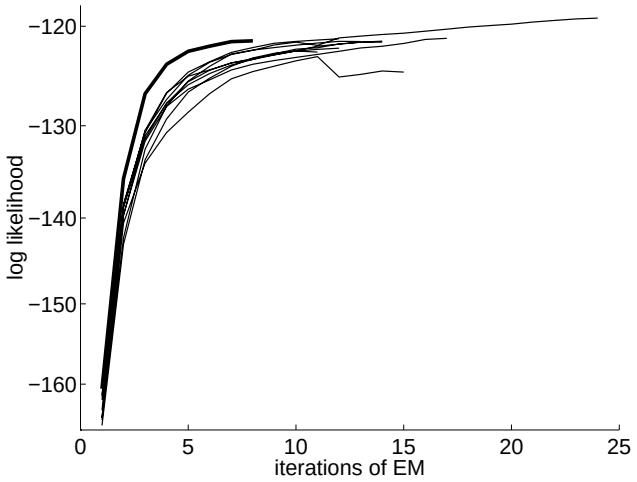
Figure 6.2 shows the results of learning a 3-cluster model, each cluster with 12 parts. Figure 6.2(a) is of single-subject models, and (b) is of multiple-people models. We used random initializations, and the ten curves in Figure 6.2(a) or (b) correspond to ten such random initializations. If the likelihood difference of two iterations is less than 0.1%, a convergence is claimed and the algorithm terminates. From Figure 6.2 we can see that generally the log-likelihoods grow and converge well with the iterations of EM.

6.4.2 Models obtained

Figure 6.2 shows that the models obtained from different initializations are quite similar by likelihood criterion. We tested the models using a small validation set, and found no big difference in terms of detection performance. Figures 6.3 (a) and (b) show a single-subject model (corresponding to the thick curve in Figure 6.2 (a)). Figure 6.3(a) 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. Figure 6.3(b) depicts the learned decomposable triangulated probabilistic struc-



(a)



(b)

Figure 6.2: Evaluation of the EM-like algorithm: log-likelihood vs. iterations of EM for different random initializations. The indices along x-axis show the number of iterations passed. (a). 12-part 3-cluster single-subject models; (b). 12-part 3-cluster multiple-people models.

ture for the three component models in (a), respectively. The letter labels show the body parts correspondence. Figure 6.3 (c) and (d) are a multiple-people model (corresponding to the thick curve in Figure 6.2 (b)), and follow the same representation custom as in (a) and (b).

6.4.3 Detection and labeling

We run detection and labeling (section 6.3) experiments using the models obtained. Instead of a fixed threshold, we represent the detection performance by receiver operating characteristics (ROC) curves. Figure 6.4 shows some ROC curves using the single-subject model as in Figure 6.3(a)-(b). Figure 6.4(a) is the ROC curves of positive walking sequences (type p1 to p3) vs. person biking R-L sequences (b+), and (b) is the ROC curves of positive walking sequences vs. car moving R-L sequences (c+). The positive examples for the solid curves are the positive R-L walking sequences (type p1 to p3) of subject LG excluding the sequences used for training, and the positive examples for the dashed curves are the R-L walking sequences (type p1 to p3) of other subjects not in the training set. From Figure 6.4 (a) and (b), we see that the single-subject model performs similarly well on the in-training-set subject and the out-of-training-set subjects. To further test if the single-subject model can distinguish the in-training-set subject from other subjects, we obtain an ROC curve (Figure 6.4 (c)) by taking the R-L walking sequences of subject LG as positive examples and the R-L walking sequences of other subjects as negative examples. From Figure 6.4, we see that it is hard to distinguish the in-training-set subject from other subjects using the single-subject model as in Figure 6.3(a). In other words, the model is invariant with respect to the subject being observed.

From an ROC curve, we can take the detection rate when $P_{detection} = 1 - P_{falsealarm}$ as an indicator of detection performance. Figure 6.5 summarizes such detection rates of positive R-L walking sequences vs. different types of negative sequences. The x-axis of Figure 6.5 displays the different types of negative examples (as described in Table 1). We first get the detection rate of each positive R-L walking sequence vs.

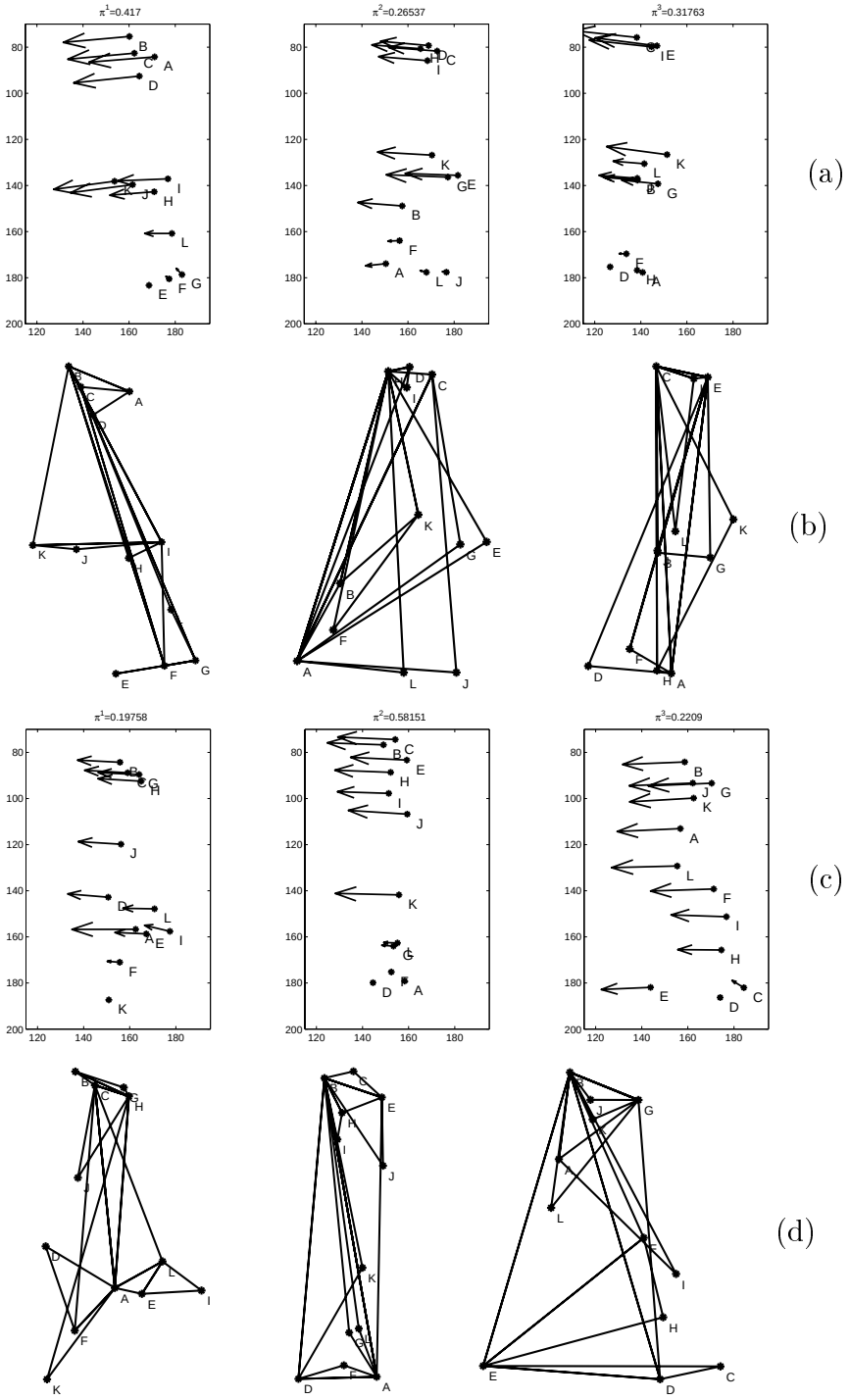
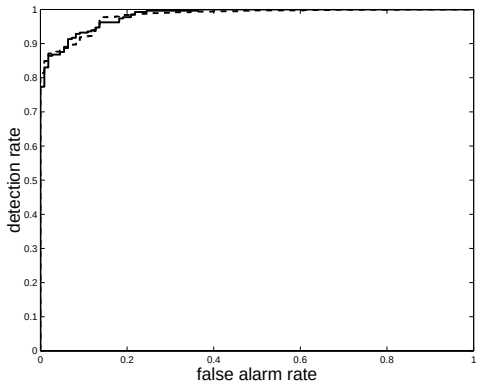
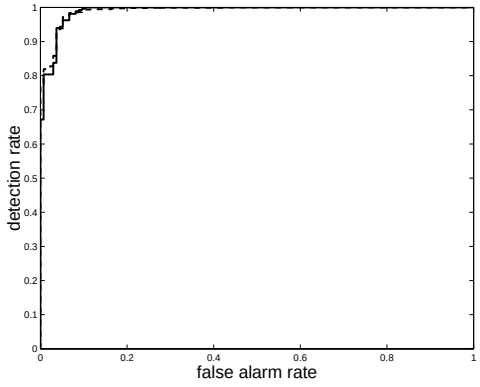


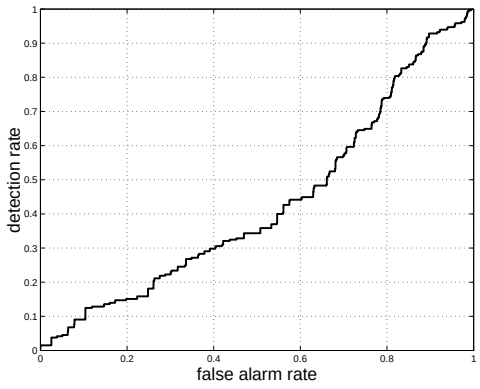
Figure 6.3: Examples of 12-part 3-cluster models. (a)-(b) are a single-subject model (corresponding to the thick curve in Figure 6.2 (a)), and (c)-(d) are a multiple-people model (corresponding to the thick curve in Figure 6.2 (b)). (a) (or (c)) gives the mean positions and mean velocities (shown in 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)



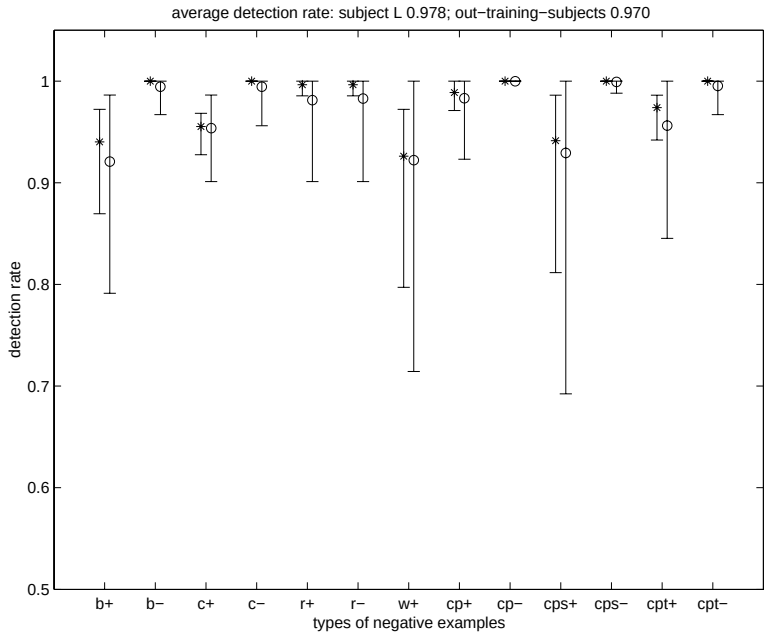
(c)

Figure 6.4: ROC curves using the single-subject model as in Figure 6.3 (a). (a) positive walking sequences vs. person biking R-L sequences (b+); (b) positive walking sequences vs. car moving R-L sequences (c+). Solid curves use positive walking sequences of subject LG as positive examples, and dashed curves use sequences of other subjects. (c) is obtained by taking the R-L walking sequences of subject LG as positive examples and the R-L walking sequences of other subjects as negative examples.

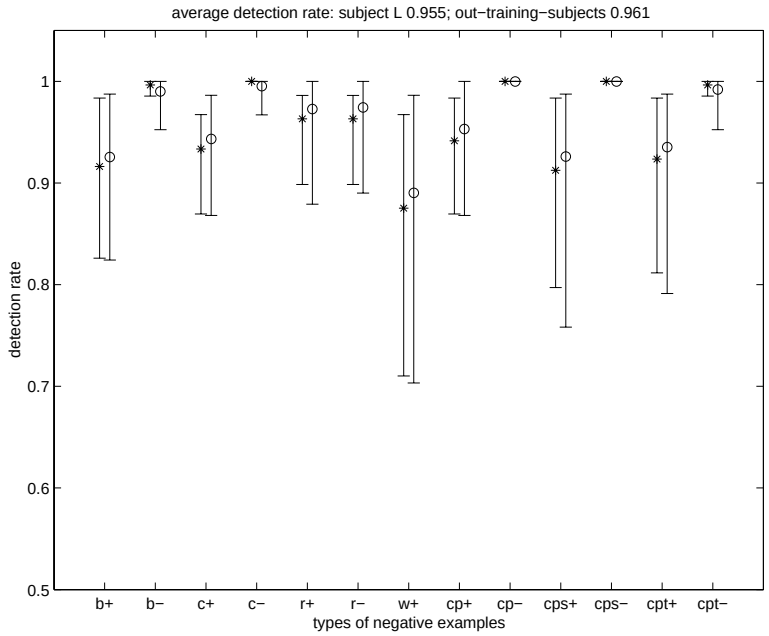
a certain type of negative sequences, and the average detection rate is shown either in star (*) or in circle (o). The error bars show the maximum or minimum detection rate. The stars (*) with error bars use the positive walking sequences of subject LG as positive examples, and the circles (o) with error bars use the positive sequences of other subjects not in the training set. Figure 6.5(a) is from the single-subject model as in Figure 6.3(a), and Figure 6.5(b) is from the multiple-people model as in Figure 6.3(c).

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 a 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 Figure 6.5, we see that the two models perform similarly, with overall detection rates (out-of-training-set subjects) of 97.0% and 96.1% for the single-subject model and multiple-people model, respectively.

Figure 6.6 shows results on some images using the 12-part 3-cluster multiple-people model (Figure 6.3 (c)). 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 6.3). 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 Figure 6.3 (c). The number at the upper right corner shows the highest likelihood component, with $1, 2, 3$ corresponding to the three components



(a)



(b)

Figure 6.5: Detection rates vs. types of negative examples. (a) is from the single-subject model (Figure 6.3 (a)), and (b) is from the multiple-people model (Figure 6.3 (b)). 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. The performance is measured on pairs of frames. It improves further when multiple pairs in a sequence are considered.

in Figure 6.3 (c) from left to right. For the samples in Figure 6.6, all the positive R-L walking examples are correctly detected, and only one negative example (from the water running R-L sequence) is wrongly claimed as a person R-L walking (a false alarm).

6.5 Conclusions

The algorithms developed in previous chapters are extended to mixtures of Gaussian with decomposable triangulated independence. We explore the efficiency and effectiveness of this 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 non-walking motions. We find an average of 4% error rate on our examples. This rate is based on pairs of frames, and it can be further improved when more pairs of frames (150-200 ms of video) are included (sections 3.3 and 3.5.2).

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

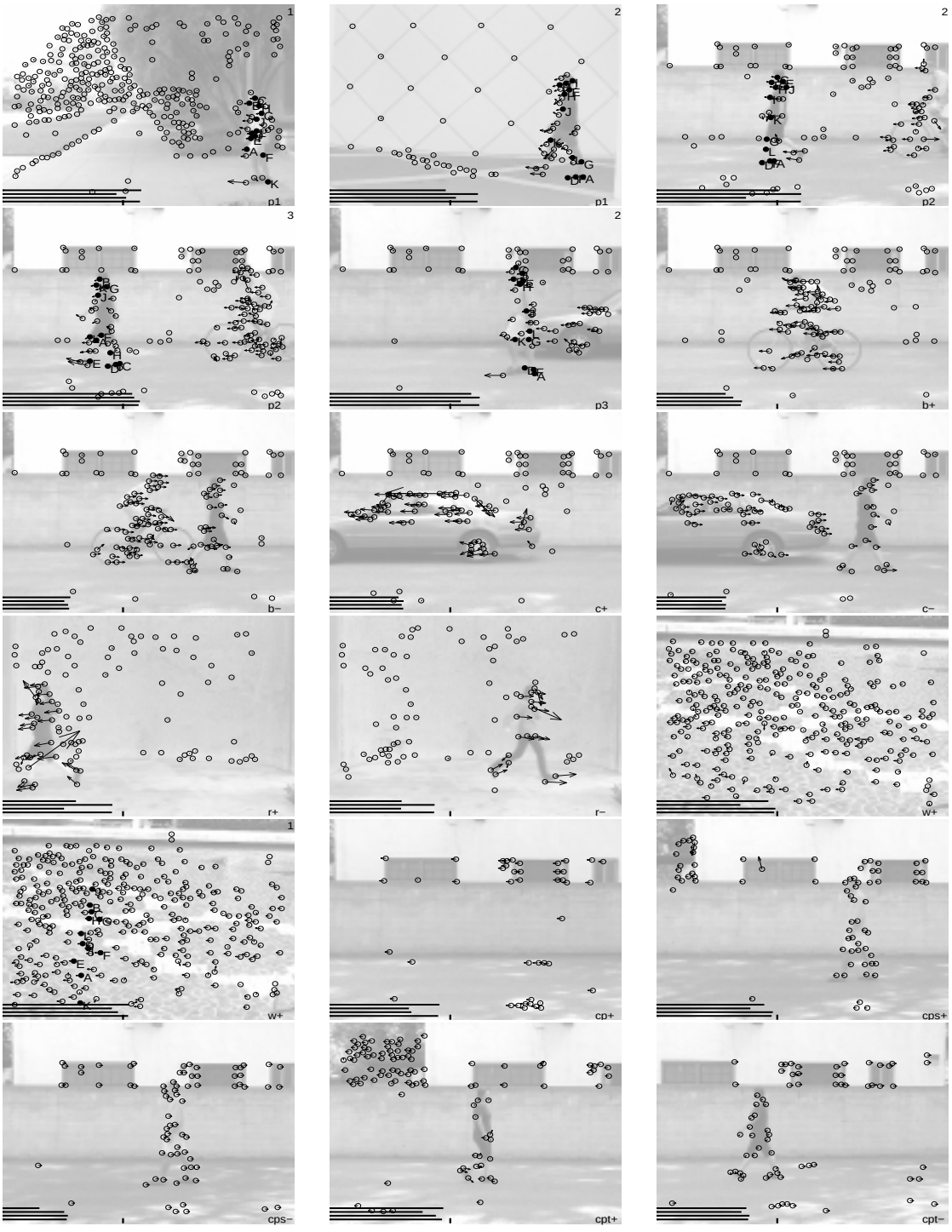


Figure 6.6: Detection and labeling results on some images. See text for detailed explanation of symbols.

Chapter 7 Decomposable triangulated graphs and junction trees

In this thesis, we use decomposable triangulated graphs to depict the probabilistic conditional independence structure of body parts. Detection and labeling can be done efficiently through dynamic programming. In this chapter, we show how a decomposable triangulated graph can be transformed into a junction tree such that max-propagation developed for graphical models can be used to the labeling problem. We also justify our choice of decomposable triangulated graphs over other types of graphical models.

This chapter does not intend to give a thorough survey of graphical models and inference algorithms. Instead we want to show how those algorithms are related to our problem.

7.1 Introduction

Graphical models are graphs which describe the probabilistic conditional (in)dependence of variables. Each node of the graph represents a random variable, and edges give the dependency among these variables. If each variable can take values from a discrete set, the configuration that maximizes the joint probability can be found efficiently by max-propagation on junction trees, which are graphs built on original graphs to make the description of inference algorithms easier.

For our labeling problem, each body part is denoted by a node in the graph, and it can take values from a set of candidate features. Therefore the labeling problem is the most-probable-configuration problem on the graph.

In the following sections, we will describe what junction trees are and how the most-probable-configuration problem is solved through max-propagation on junc-

tion trees. We will compare the dynamic programming algorithm with the max-propagation on junction trees and finally justify our choice of decomposable triangulated graphs from a graphic theoretical point of view.

7.2 Junction trees

A *clique tree* is a tree in which the nodes are the cliques of an underlying graph. For example, Figure 7.1 shows examples of clique trees. The edges of a clique tree

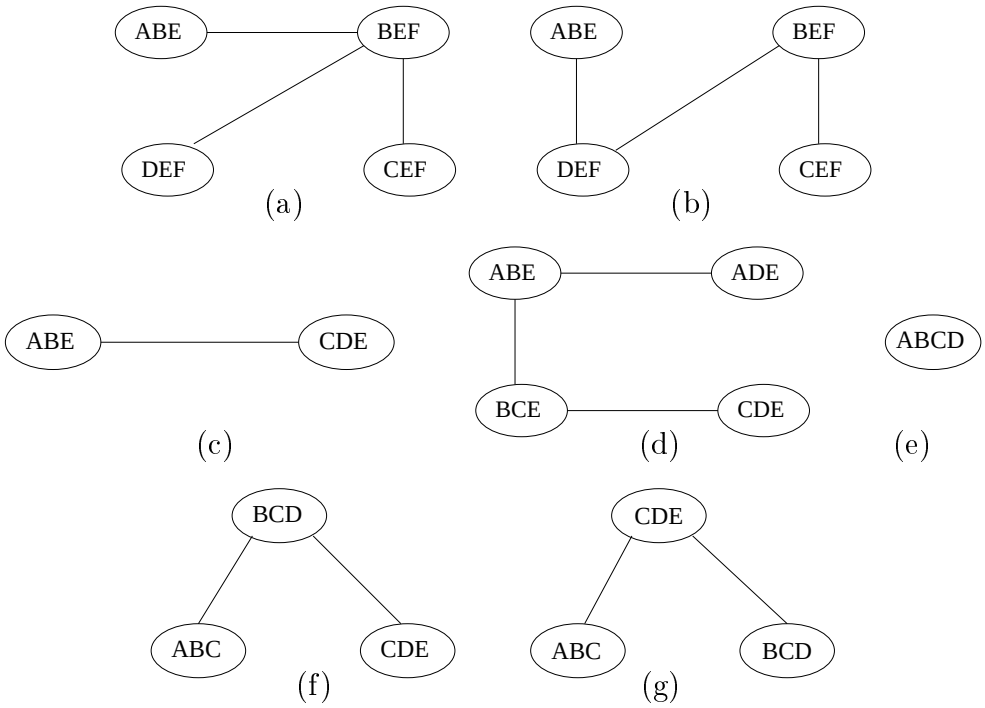


Figure 7.1: Examples of clique trees. (a) and (b) are for the graph in Figure 2.2; (c), (d) and (e) are for the graphs of Figure 2.4 (a,b,c), respectively; (f) and (g) are for the graph in Figure 2.5. (a,c,e,f) are junction trees, and (b,d,g) are not.

can be labeled with *separators*-the intersection of the corresponding cliques of the two adjacent nodes. Figure 7.2 shows some cliques trees with separators. A *junction tree* is a clique tree with the property that the nodes containing the same variable are connected, which is called 'junction tree property'. Not all the clique trees are junction trees. In Figure 7.1, (a,c,e,f) are junction trees, and (b,d,g) are not. All the graphs have clique trees, but not all the graphs have junction trees. For example,

there is no junction tree for the graph of Figure 2.4(b). There exists a junction tree if and only if a graph is triangulated or decomposable. A graph is triangulated if there are no chordless cycles in the graph. A graph is decomposable if there exists an elimination order of the vertices such that when a vertex is eliminated, all the vertices connected to it are connected with each other. It can proved that triangulated and decomposable are two equivalent properties. A non-triangulated graph can become triangulated by adding edges.

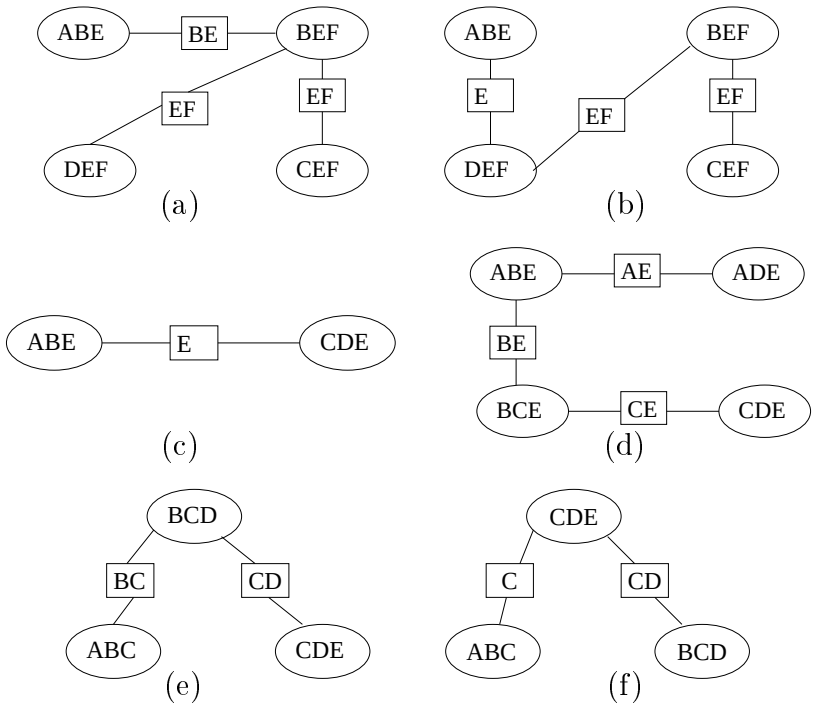


Figure 7.2: Examples of clique trees with separators. Clique trees are from Figure 7.1.

For the decomposable triangulated graphs (see notes in section 2.2) we used in the previous chapters, all the cliques are of size three, and all the separators are of size two. Figure 7.3 shows one junction tree for Figure 2.3 (a).

7.3 Max-propagation on junction trees

Let U be the set of variables. We consider a clique tree over U. Let $\mathcal{C}$ denote the set of cliques, and $\mathcal{S}$ denote the set of separators. We define *potentials* $\Psi_C(X_C)$ on

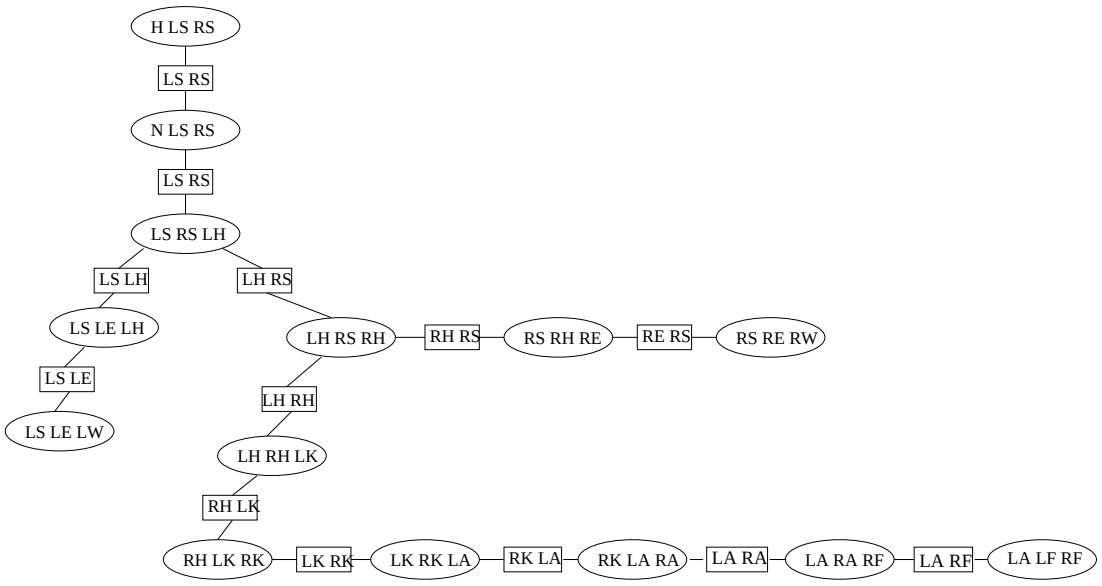


Figure 7.3: A junction tree with separators for the body decomposition graph in Figure 2.3 (a).

each clique $C \in \mathcal{C}$, and $\Phi_S(X_S)$ for each separator $S \in \mathcal{S}$. Potentials are non-negative functions, and in order to make the max-propagation work, the only condition on the initial potentials is that the joint probability can be expressed as

$$P(X_U) = \frac{\prod_{C \in \mathcal{C}} \Psi_C(X_C)}{\prod_{S \in \mathcal{S}} \Phi_S(X_S)} \tag{7.1}$$

For example, we can initialize $\Psi_C(X_C) = P(X_C)$ and $\Phi_S(X_S) = P(X_S)$.

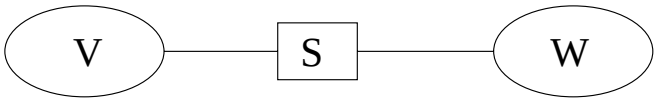


Figure 7.4: Two cliques V and W with separator S.

Let us first consider max-propagation on two cliques. Figure 7.4 shows cliques V

and W with separator S. If we want to pass message from clique V to clique W, then

$$\Phi_S^* = \max_{V \setminus S} \Psi_V \tag{7.2}$$

$$\Psi_W^* = \frac{\Phi_S^*}{\Phi_S} \Psi_W, \tag{7.3}$$

where the asterisk means the updated value. It can be easily proved that the joint probability remains unaltered after this message passing, that is, if we define $\Psi_V^* = \Psi_V$, then $\frac{\Psi_V^* \Psi_W^*}{\Phi_S^*} = \frac{\Psi_V \Psi_W}{\Phi_S}$. Similarly, if we want to pass message from W to V,

$$\Phi_S^{**} = \max_{W \setminus S} \Psi_W^* \tag{7.4}$$

$$\Psi_V^{**} = \frac{\Phi_S^{**}}{\Phi_S^*} \Psi_V^*. \tag{7.5}$$

Again the joint probability remains unchanged. Another important property is that after one pair of message passing, local consistency holds, which is

$$\max_{V \setminus S} \Psi_V^{**} = \max_{W \setminus S} \Psi_W^{**} \tag{7.6}$$

For a clique tree, if we pass the messages according to the *Message-Passing protocol*: a clique can send a message to a neighboring clique only when it has received messages from all of its other neighbors, then after one round of message passing, local consistency between any two cliques are guaranteed. For a junction tree, local consistency implies global consistency because in a junction tree, if some variables are common for two cliques, they are common for all the cliques on the path of those two cliques. More importantly, the junction tree property ensures that when the message passing terminates, we have

$$\Psi_C(X_C) = \max_{U \setminus C} P(X_U) \tag{7.7}$$

From equation (7.7), the most probable configuration can be found through maximization over each individual clique. If for one clique there are multiple max config-

urations, then we can choose one of the configurations, and take it as evidence to run max-propagation again until one most-probable-configuration is obtained.

Assume that M is the number of variables, N is the number of the values each variable can take, and r is the maximum clique size. From equations (7.2) and (7.3), the cost of one message passing is on the order of M^r, exponential with the maximum clique size r. If the underlying graph is connected, the number of cliques in a junction tree is no more than $N-1$ and the number of edges (separators) is no more than $N-2$. Hence in one run of the above max-propagation, there are at most $2(N-2)$ message passing. For decomposable triangulated graphs, there are $N-2$ cliques and $2(N-3)$ message passing.

7.4 Comparison between dynamic programming and max-propagation on junction trees

The above max-propagation algorithm can work on any junction trees, so the labeling problem can be solved on graphs other than decomposable triangulated graphs. Dynamic programming can also work on graphs other than decomposable triangulated graphs ([32]). The computational complexity of both algorithms are determined by the maximum size of the cliques. Therefore they essentially have the same order of complexity. But there are still differences. Dynamic programming is an elimination algorithm ([33]) for the most-probable-configuration problem. For general marginalization problems, probability propagation on junction trees is more efficient (better) than an elimination algorithm, because we can obtain all the needed marginal probabilities after one round of probability propagation, and for the elimination algorithm we may need to run the algorithm many times. But for our labeling (most-probable-configuration) problem, dynamic programming is designed directly to solve it and is more efficient than max-propagation on junction trees. The cost of dynamic programming is about one-half of that of one round max-propagation since the computational cost of one stage of dynamic programming (section 2.3) is about the same as a single

message passing from one clique to another clique (equations (7.2,7.3)).

Note that the max-propagation algorithm on junction trees described above is just one inference algorithm (Hugin algorithm [34, 33]) on graphical models. There exist other inferences algorithms. But they are essentially the same.

7.5 Justification for the use of decomposable triangulated graphs

The computational complexity of both algorithms is determined by the maximum size of the cliques. Therefore, the type of graph we used, i.e, decomposable triangulated graphs, is the most powerful one among the graphs with similar computational cost. It is the most powerful in the sense that it can model any probabilistic (in)dependency that can be modeled by a graph with maximum clique size three. This is because for any decomposable graph with maximum clique size three, we can always convert it into a decomposable triangulated graph by adding edges. By the same reasoning, decomposable triangulated graphs are more powerful than the decomposable graph with maximum clique size of less than three, e.g., trees. In other words, for any probability distribution, decomposable graphs can provide the most accurate approximation among all the decomposable graphs with maximum clique size equal to or less than three. The family of probability distribution represented by decomposable triangulated graphs is the same as the family represented by all the decomposable graph with maximum clique size equal to or less than three.

From the above discussion, we know 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 proved that the latter problem is NP-hard ([35] and [36]). Therefore, the search of optimal decomposable triangulated graph is NP-hard.

The method of transforming a decomposable graph with maximum clique size equal to or less than three into a decomposable triangulated graph is quite straightforward. First we need an elimination ordering of vertices. When we subsequently

delete a vertex, we ensure that it is contained in one triangle by adding edges. Figure 4.1 in Chapter 4 shows an example of transforming a tree into a decomposable triangulated graph.

7.5.1 Trees vs. decomposable triangulated graphs

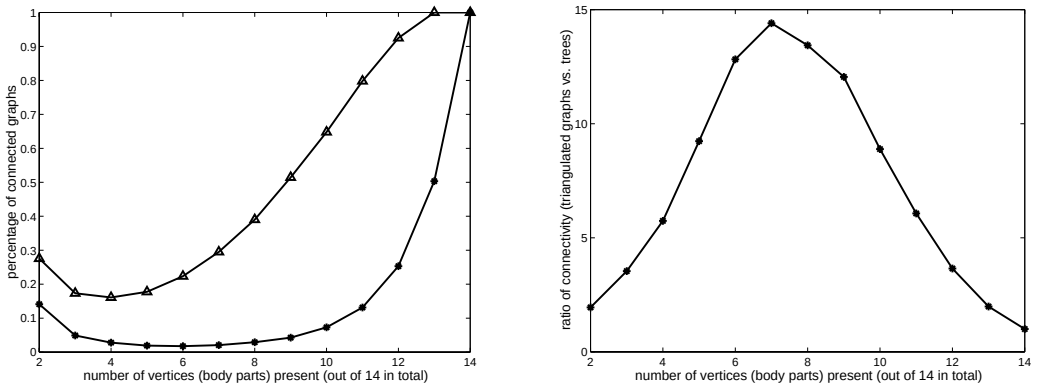
Trees are a type of widely studied and used graphs. Trees have computational advantages over decomposable triangulated graphs. First, the maximum clique size of a tree is two, so trees have a lower computational cost. Second, there exist efficient algorithms that can guarantee the finding of the optimal tree (max-spanning-tree algorithms [26]). But the following advantages of decomposable triangulated graphs make them a better choice for our problem.

Power of the model. In section 4.3, we give an algorithm for finding decomposable triangulated dependency based on trees. We first find the best tree dependency by the maximum spanning tree algorithm, and then transform it into a decomposable triangulated model by adding edges. This method guarantees that the decomposable graph obtained is not worse than the optimal tree for any given optimization criterion (mutual information for our problem), because the set of probability independences described by trees is a subset of that by decomposable triangulated graphs.

Graph connectivity in case of occlusion. A connected graph means that there is a path between any two vertices of a graph. Graph connectivity is very important for our problem. The body parts have only the local dependence described by the graph. If some body parts are missing (occlusion), other body parts can become disconnected, and therefore independent. Then the graph is no longer a good approximation of the true probability density function. For example, in Figure 4.1, if vertex B is missing, then the tree in Figure 4.1(a) becomes four mutual independent subgraphs, (A), (C, F, G), (D, H, I, J) and (E), but the decomposable graph in Figure 4.1(b) is still a connected graph. In fact, it is generally true that decomposable triangulated graphs have advantages over trees on the property of graph connectivity.

If the number of vertices is M, then the number of edges in a tree is $M - 1$, and

it is $2 * M - 3$ for a decomposable triangulated graph. A decomposable graph has almost twice as many edges as a tree. But the advantage on connectivity is more than twice. We test the connectivity under occlusion on a tree (solid lines in Figure 4.2(c)) and a decomposable triangulated graph (solid and dashed lines in Figure 4.2(c)). We randomly select some body parts and check if they belong to a connected subgraph. For example, (LE, LS, LH) is connected in both the tree and the triangulated graph, but (LE, LS, N) is connected in the triangulated graph but not in the tree. We run the algorithm many times with random selection of body parts. Figure 7.5 shows the results. Figure 7.5 (a) shows the percentage of connected graphs vs. number of vertices present (out of 14 in total). The solid line with stars is for the tree, and the line with triangles for the decomposable triangulated graph. Figure 7.5 (b) gives the ratio of the connected percentage as in Figure 7.5 (a). The connected percentage of the decomposable triangulated graph is divided by that of the tree. The average ratio considering all the situations is more than 7. When we average over the cases of four to twelve body parts present, which is more representative of typical situations of real occlusion, the ratio is around 10.



(a) (b)

Figure 7.5: (a) percentage of connected graphs vs. number of vertices present (out of 14). The solid line with stars is for the tree, and the line with triangles for the decomposable triangulated graph. (b) the ratio of connected percentage: decomposable triangulated graphs vs. trees.

Figure 7.5 gives an idea of how connectivity changes with the number of vertices present in the general case. It is not a very accurate model for the case of human

motion since occlusion doesn't happen completely randomly. For example, the left wrist and elbow may disappear or appear together, and the right wrist and elbow may disappear or appear together. This correlation of the body part presence might be used to construct graphs with better connectivities.

In case of occlusion we can use the approximation described in chapter 3 to deal with missing body parts. But for trees the problem becomes harder because once a non-leaf node is missing, the original tree becomes several isolated subgraphs. Those subgraphs are independent of each other, which makes it difficult to find a connected human body.

Translation invariance As we mentioned in the previous chapters (sections 2.3 and 4.4), we use local coordinate systems to deal with translation invariance. For example, for a triangle (A_t, B_t, C_t), we take one body part (for example B_t) as the origin, and use relative positions for other body parts. When we compute conditional probability $P(A_t|B_t, C_t)$, there can be relative position information on both sides of the condition sign (|), but for trees the position information can only be on one side, which makes the model less descriptive.

The weakness of trees may be compensated by a mixture of trees ([21, 37]). In [37], rectangular segments instead of point features are used as body parts so that one edge of the tree in [37] contains a similar amount of human body information as one triangle in our model and therefore translation invariance can be achieved (but the connectivity is still that of a tree).

7.6 Summary

Graphical models are a very active research area, and show great promise for computer vision problems. In this chapter, we place our decomposable triangulated model in the general framework of graphical models, and show how the labeling problem can be solved by max-propagation on junction trees. The dynamic programming algorithm is compared with the inference algorithms of graphical models.

Decomposable triangulated graphs are the most powerful among graphs with sim-

ilar or less computational cost for inference. We justify our choice of decomposable triangulated graphs over trees by the accuracy of the model, graph connectivity in case of occlusion, and the ability to achieve translation invariance.

Chapter 8 Conclusions and future work

In this thesis, we have presented a probabilistic approach to human motion detection and labeling, i.e., how to perform human motion detection and labeling using a probabilistic model and how to learn the probabilistic model from data. Section 8.1 summarizes the main contributions of this thesis. Section 8.2 outlines future directions for improving and generalizing this work.

8.1 Summary of main contributions

In the field of computer vision, human motion detection is a very important problem, but it has never been tackled before due to its inherent difficulty. This thesis proposes a learning based probabilistic approach to solve the problem. Under the assumption that the human body is composed of body parts, graphical models are originally deployed to model the joint probability density function (PDF) of the position and velocity of the body parts so that a combinatorial search is avoided and detection and labeling are performed efficiently. The proposed method can handle occlusion and extraneous clutter in a systematic way - a challenging scenario for computer vision algorithms.

This thesis also makes important contributions to the learning of graphical models. An unsupervised learning algorithm that can obtain a probabilistic model of an object, independence structures, as well as model parameters automatically from unlabeled training data has been presented. It is the first work that can learn graph structure from training data including useful foreground features and irrelevant background clutter with unknown correspondence between the parts and the features. Model learning can also be performed when features belonging to some foreground parts are missing (occlusion). This algorithm enables the creation of systems that are able to learn models of human motion (or other objects) completely automatically from real

image sequences.

All the algorithms presented in this thesis are tested and supported by experiments on motion capture data and/or grayscale image sequences. When we learn a mixture model of right-to-left walking and test on walking sequences of a number of people as well as a variety of non-walking motions, detection rates of over 95% are achieved on pairs of frames.

8.2 Future work

We have demonstrated the efficiency and effectiveness of our algorithms through experiments on learning and testing a model of side-view walking. This work can be extended, improved and further experimented in the following aspects.

The algorithms can be extended to other types of graphical models, since both the labeling algorithm and the unsupervised technique are not limited to decomposable triangulated graphs. Loopy graphical models may be used to more accurately model the conditional independence of variables and provide hopes for handling occlusion in a more precise way, i.e., without stronger independence assumption. This work has already been started in the vision lab at Caltech.

In this thesis we used measurements from pairs of frames in most experiments. Considering more frames can improve the detection performance (section 3.5.2). There are other ideas which can make use of the information from more frames (longer duration). A straightforward way is to model the whole body using a higher dimensional Gaussian to include measurements from multiple frames. Another more sensible way in dealing with more complex motion than walking is to add a higher level temporal correlation (dynamics) among frames. For example, the models described in this thesis can be applied on pairs of frames, and the relation among the pairs of frames can be captured using the idea of Hidden Markov Model ([38]).

There is also more work to be done on the experimental side. (1) Experiments can be conducted with different types of motion beyond walking. (2) The trade-off between model complexity (number of clusters and number of parts) and accuracy

should be systematically studied. (3) Viewpoint invariance and scale invariance can be studied on grayscale image sequences. (4) The algorithm can be made to run faster on images with a large number of features. One possible way to speed up is to build a pyramid-like hierarchy system. Another idea is to decompose an image into several subregions and run the algorithm on each subregion. (5) Other types of features, including other point feature detectors/trackers or non-point features, may also be examined to see if the system can be further improved.

Bibliography

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

[2] S. Soatto, *A Geometric Framework for Dynamic Vision*, Ph.d. thesis, Caltech, 1996.

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

[4] N. Howe, M. Leventon, and W. Freeman, "Bayesian reconstruction of 3d human motion from single-camera video", *Tech. Rep. TR-99-37, a Mitsubishi Electric Research Lab*, 1999.

[5] 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.

[6] J.M. 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, 1994.

[7] 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.

[8] 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.

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

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

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

[12] R. Polana and R. C. Nelson, "Detecting activities", in *DARPA93*, pages 569–574, 1993.

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

[14] G. Johansson, "Visual perception of biological motion and a model for its analysis", *Perception and Psychophysics*, 14:201–211, 1973.

[15] J.E. Cutting and L.T. Kozlowski, "Recognizing friends by their walk: Gait perception without familiarity cues", *Bulletin Psychonometric Society*, 9:353–356, 1977.

[16] G. Mather and L. Murdoch, "Gender discrimination in biological motion displays based on dynamic cues", *Proc. R. Soc. Lond. B*, 259:273–279, 1994.

[17] W. Dittrich, T. Troscianko, S. Lea, and D. Morgan, "Perception of emotion from dynamic point-light displays represented in dance", *Perception*, 25:727–738, 1996.

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

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

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

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

[22] M. Weber, M. Welling, and P. Perona, "Unsupervised learning of models for recognition", in *Proc. ECCV*, volume 1, pages 18–32, 2000.

[23] P. Neri, M.C. Morrone, and D.C. Burr, "Seeing biological motion", *Nature*, 395:894–896, 1998.

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

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

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

[27] M. Weber, *Unsupervised Learning of Models for Object Recognition*, Ph.d. thesis, Caltech, 2000.

[28] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the em algorithm", *J.R.Statist. Soc. B*, pages 39:1–38, 1977.

[29] M. Welling, "Em-algorithm", Technical report, class notes at California Institute of Technology, 2000.

[30] C. M. Bishop, *Neural networks for pattern recognition*, Clarendon Press, 1995.

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

[32] U. Bertele and F. Brioschi, *Nonserial dynamic programming*, Academic Press, 1971.

[33] M. I. Jordan and C. M. Bishop, *Introduction to Graphical Models*, unknown, soon.

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

[35] D. Chickering, D. Geiger, and D. Heckerman, "Learning bayesian networks is np-hard", Technical report, Microsoft Research, MSR-TR-94-17, 1994.

[36] N. Srebro, "Maximum likelihood bounded tree-width markov networks", in *UAI*, pages 504–511, San Francisco, CA, 2001.

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

[38] L. Rabiner and B. Juang, *Fundamentals of speech recognition*, Prentice Hall, 1993.