NORTHWESTERN UNIVERSITY

Probabilistic Variational Methods for Vision based Complex Motion Analysis

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY

Field of Electrical and Computer Engineering

By

Gang Hua

EVANSTON, ILLINOIS

June 2006

© Copyright by Gang Hua 2006

All Rights Reserved

ABSTRACT

Probabilistic Variational Methods for Vision based Complex Motion Analysis

Gang Hua

Many emerging applications, including intelligent vision based human computer interaction, intelligent video surveillance, virtual and augmented reality, animation, and biomedical image analysis for computer aided diagnosis and surgery, demand effective and efficient vision-based methods to analyze complex motions, such as articulated motion, deformable motion, and multiple motions. The fundamental challenges of this inverse problem come from two aspects: the high degrees of freedom in these complex motions, and the complications in the image measurements. The high dimensionality of this problem has plagued the scalability and efficiency of many existing methods.

In search for a new and scalable solution that overcomes the curse of dimensionality, we view this problem from another angle and conjecture that the complexity of such a problem can be approached by the collaboration among a set of low dimensional motion estimators. Targeting on the two fundamental challenges, in this dissertation, we propose a distributed and collaborative probabilistic reasoning framework for complex motion analysis. The theoretic foundation of the proposed approach is based on probabilistic graphical models. The probabilistic variational inference on these graphical models clearly reveals a set of simple

interactive Bayesian motion estimators, which obtains the optimal solution in a collaborative fashion. All computations in the Bayesian inference can be efficiently performed in a distributed and parallel way.

The proposed collaborative approach is also a distributed visual measurement integration framework, which handles *measurement uncertainty*, *measurement multi-modality*, and *measurement inconsistency* in a principled way. Extensive experimental results on analyzing different complex motions demonstrate the effectiveness, efficiency, scalability and robustness of the proposed collaborative approach.

To my parents, my sisters, my brothers, and Yan

Acknowledgments

Words can not express my greatest gratitude to my advisor, Prof. Ying Wu for his unreserved support, elaborate guidance, and inspiring encouragement throughout my Ph.D. study. What I learned from him would be always beneficial for my future career.

I would express my sincere appreciation to my mentors at Microsoft Research (MSR), Dr. Zicheng Liu and Dr. Zhengyou Zhang, for their selfless advice and guidance, which greatly broaden my horizon. The same appreciation is expressed to Dr. Ming-Hsuan Yang for his suggestions during and after my summer internship at Honda Research Institute (HRI), which greatly improve part of this work. I would also like to thank my Ph.D. committee members, Prof. Aggelos K. Katsaggelos and Prof. Thrasyvoulos N. Pappas., for their constructive discussions and comments.

Many thanks to my colleagues at Northwestern University, as well as many other friends at MSR and HRI. In particular, I would like to thank Dr. Jasha Droppo, Dr. Asela Gunawardana, Dr. Mike Seltzer, Dr. Li Deng, Ting Yu, Ming Yang, Junsong Yuan, Shenyang Dai, Zhimin Fan, Dr. Junqing Chen, Ning Wen, Dr. Xiaolong Li, Ming Liu, Aravind Sundaresan, Vasco Pedro, whose friendship would be a lifetime fortune for me.

I am always cordially thankful to my parents, my sisters and my brothers, whose endless love and unconditional support accompanied with me throughout all these years of my study abroad. Last but by no means least, I owe my wife Yan Gao, for her love, understanding, support, sacrifice and help with all her heart, which make this work possible.

Contents

ABSTRACT 3

Acknowledgments 6

List of Tables 12

List of Figures 13

Chapter 1. Introduction 16

1.1. Background 16

1.1.1. Visual analysis of complex motion 16

1.1.2. Visual measurement integration 18

1.1.3. Distributed and parallel computing 19

1.1.4. Probabilistic reasoning and graphical models 21

1.2. Motivation 22

1.3. Organization 23

1.4. Contributions 24

Chapter 2. Probabilistic inference on graphical models: a brief review 26

2.1. Introduction 26

2.2. Basic problems of probabilistic inference 28

2.2.1. Latent variable inference 29

2.2.2. Parameter estimation 30

2.2.3. Model selection and model averaging 31
2.3. Graphical model representation 33
2.3.1. Belief networks, Bayesian network, and dynamic Bayesian network 33
2.3.2. Markov random field and Markov network 34
2.3.3. Factor graph 37
2.3.4. Equivalence of the different graphical models 38
2.4. Probabilistic inference algorithm 38
2.4.1. Exact inference algorithm 38
2.4.2. The variational approach 41
2.4.3. Monte Carlo methods 47
2.4.4. Miscellaneousness 52
2.4.5. Model selection and model scoring criteria 52
2.5. Conclusion remarks 54

Chapter 3. Mean field variational analysis for articulated body tracking 55
3.1. Introduction 55
3.2. Related work 57
3.3. The representation of an articulated body 59
3.4. Mean field variational analysis 62
3.5. Mean field Monte Carlo (MFMC) 65
3.6. Dynamic Markov network and sequential mean field Monte Carlo 68
3.7. Experiments on tracking articulated body 71
3.7.1. Experimental setup 71
3.7.2. Results of MFMC iteration 72

3.7.3. Various articulated objects 73
3.8. Conclusion 75

Chapter 4. Variational maximum a posterior estimation 77
4.1. Introduction 77
4.2. Related work 79
4.3. Kullback-Leibler divergence between a Gaussian and an arbitrary p.d.f. . . . 82
4.4. Gaussian mean field variational analysis 84
4.5. Variational MAP by deterministic annealing 89
4.6. Monte Carlo simulation of the variational MAP 90
4.7. Validation experiments and application to articulated body tracking 92
4.7.1. Evolution of the topology of the KL divergence during annealing 94
4.7.2. Variational MAP inference in an illustrative synthetic problem 94
4.7.3. Variational MAP for tracking articulated human body 100
4.8. Conclusion and future work 106

Chapter 5. Data driven belief propagation for human pose estimation 108
5.1. Introduction 108
5.2. Prior work and context 110
5.3. Bayesian formulation 112
5.3.1. Markov network 112
5.3.2. Pose parametrization 113
5.3.3. Potential function and likelihood model 116
5.4. Data driven belief propagation 118
5.4.1. Belief propagation Monte Carlo 118

5.4.2. Data driven importance sampling 120
5.5. Experiments on human pose estimation 125
5.5.1. Validation of the likelihood model 125
5.5.2. Pose estimation results 126
5.5.3. Discussions 129
5.6. Concluding remarks 130

Chapter 6. Robust integration of inconsistent measurement 131
6.1. Introduction 131
6.2. Formulation of multi-source integration 134
6.3. Measurements inconsistency 135
6.4. Detection of inconsistency and falseness 139
6.5. Robust integration for ensemble tracking 140
6.6. Experiments 141
6.6.1. Illustrative numerical example 141
6.6.2. Robust part based ensemble tracking 143
6.7. Conclusions and future work 148

Chapter 7. Conclusion and future research 149
7.1. Summary 150
7.2. Future research 151

References 153

Appendix A. Lemmas of Theorem 4.3.1 166

Appendix B. Proof of Theorem 4.3.1 171

Appendix C. Proof of Theorem 6.3.3 173

Appendix D. Proof of Corollary 6.3.4 175

Appendix. Vita 176

List of Tables

2.1 Detailed notations. 29

3.1 A comparison of the computation of different articulated objects. The exponential requirement for computation is overcome as expected. 75

5.1 Average root mean square error (RMSE) of the estimated 2-D pose for each body part and for the whole body (e.g., LUA refers to left-upper-arm). 128

List of Figures

1.1 Possible applications of complex motion analysis. 17

2.1 A Bayesian network. 34

2.2 Neighborhood system and cliques. 35

2.3 A Markov network. 36

2.4 A factor graph. 37

2.5 Message passing example. 39

2.6 Example of junction tree. 42

3.1 The Markov network for an articulated body. 60

3.2 The constraint of two articulated parts. 61

3.3 Three factors affecting the mean field updating. 64

3.4 Importance density. 68

3.5 Dynamic Markov network for articulated body motion. 69

3.6 Iterations of mean field Monte Carlo on 2-part arm. 72

3.7 Iterations of mean field Monte Carlo on 3-part finger. 72

3.8 Tracking arm by mean field Monte Carlo. 72

3.9 Tracking arm by multiple independent tracker. 73

3.10 Tracking finger by mean field Monte Carlo. 74

3.11 Tracking upper-body by mean field Monte Carlo. 74

3.12 Tracking full-body by multiple independent tracker. 75

3.13 Tracking full-body by mean field Monte Carlo. 75

4.1 An example of Markov network. 85

4.2 Variational MAP algorithm. 90

4.3 Monte Carlo implementation of the variational MAP algorithm. 92

4.4 Evolution of the $KL(q(\mathbf{x})\|p(\mathbf{x}))$ w.r.t. μ during annealing on a synthetic example. 93

4.5 Two-nodes Markov network for the illustrative synthetic problem. 95

4.6 Convergence of the Variational MAP on the illustrative example. 97

4.7 The change of $KL(Q_1(\mathbf{x}_1)Q_2(\mathbf{x}_2\|P(\mathbf{x}_1,\mathbf{x}_2|\mathbf{z}_1 = 10.0, \mathbf{z}_2 = 16.0)))$ during annealing. 99

4.8 Annealed iteration of Eq. 4.15 in the 2-D illustrative example. 101

4.9 Tracking human body by Variational MAP. 102

4.10 More results on tracking full human body. 103

4.11 Tracking full-body motion with clutter background by variational MAP. 105

5.1 Markov network for human body pose. 113

5.2 Examples of labeled images. 114

5.3 Normalization of the labeled shape. 114

5.4 Probabilistic principal component analysis to learn the shape representation. . . 115

5.5 Definition of potential functions for body articulation. 117

5.6 Data driven belief propagation. 120

5.7 Importance functions for face, lower-arm and upper-leg. 121

5.8 Importance function for torso. 125

5.9 Importance functions for upper-arm and lower-leg. 126

5.10 Experimental results of human pose estimation. 127

5.11 Overall RMSE of each of the test images. 129

6.1 The change of σ_{ij}^2 in the Bayesian EM on an illustrative example. 142

6.2 Robust integration with flow based part measurements. 143

6.3 Typical tracking failure of Lucas-Kanade tracker and holistic particle filter. . . . 143

6.4 Comparison of the proposed robust integration with blind integration. 145

6.5 Robust integration with part measurements from particle filtering. 145

6.6 Root square errors of the results on the car sequences in the first row of Fig. 6.5 146

6.7 Tracking a group of persons by robust integration. 147

6.8 Tracking a group of cars by robust integration. 147

CHAPTER 1

Introduction

1.1. Background

1.1.1. Visual analysis of complex motion

Visual motion analysis, or visual tracking, has continued to be an active research area in computer vision for decades. The task is to recover the parameters of the target movement by analyzing the input videos or image sequences. Unlike relatively simple single motion, where only limited motion parameters need to be estimated for single object, complex motions consist of multiple correlated motions, such as the motion of articulated structure [**105**, **106**, **108**, **126**], the motion of complex deformable shapes [**18**, **19**, **133**, **141**], and the motion of multiple objects [**63**, **64**, **90**, **137**]. While it has been more or less successful to deal with simple single motion [**17**, **35**, **52**, **53**], the solutions to complex motion analysis from video are still far from satisfactory.

Effective and efficient complex motion analysis may greatly facilitate the advancement in the emerging application areas of, but are not necessarily limited to, *intelligent vision based human computer interaction* [**12**, **42**, **48**, **67**, **122**], *motion capturing for animation* [**13**, **107**], *intelligent video surveillance* [**56**, **57**, **139**, **140**], *biomedical image analysis for computer aided diagnosis and surgery* [**141**], and *humanoid robotics*, as shown in Fig. 1.1.

There are two fundamental challenges commonly reside in complex motion analysis. The first fundamental challenge is the complications of complex motions themselves. Complex motions have high degrees of freedom. The solutions to such high dimensional problems

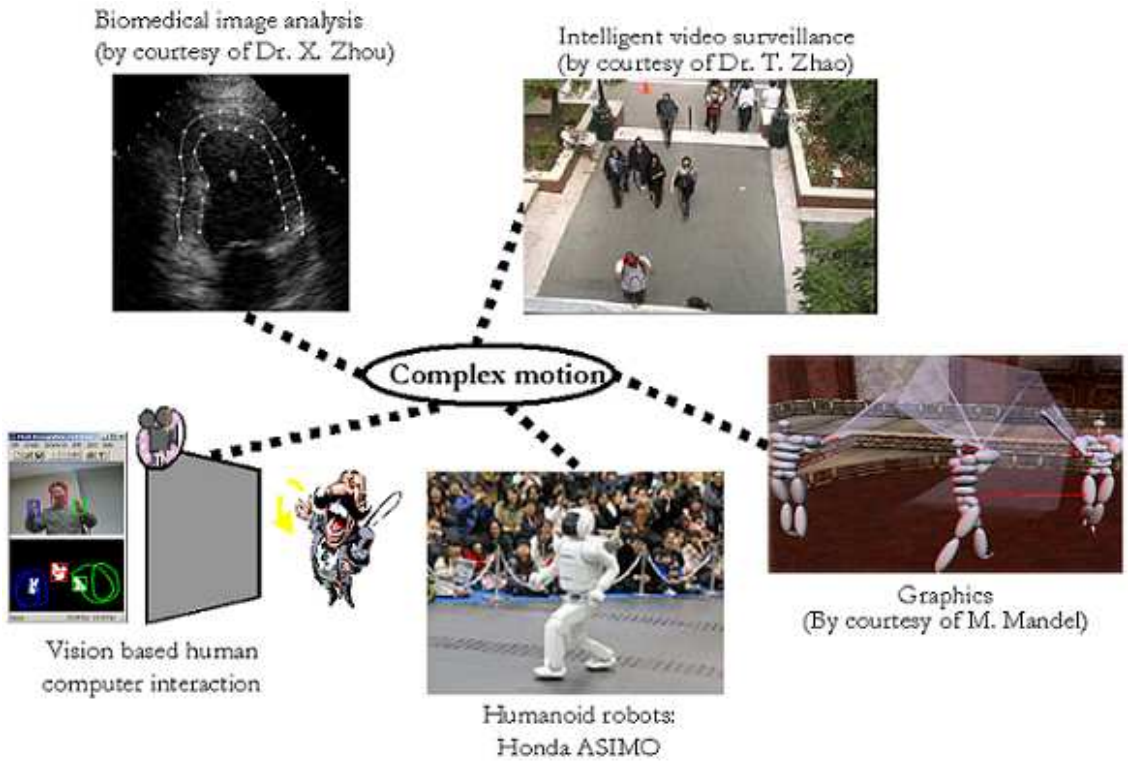


Figure 1.1. Possible applications of complex motion analysis.

potentially require tremendous computational cost. Since there are complex constraints among the multiple motions, mostly we can learn a manifold [**8**, **104**, **131**] to characterize them in a relative lower dimensional space. However, the intrinsic dimensionality of the learned manifold may still be quite high, which still hinders efficient solution.

The second fundamental challenge is the complications of image measurements. We regard the image to be jointly generated by the multiple motions and the background. Then given the image data, the background clutter may interfere with the image measurement of the complex motion. Moreover, the posterior beliefs of the multiple correlated motions are conditional dependent. Such kind of conditional dependencies also hinder efficient solution to this type of high dimensional problems.

Viewing these challenging issues from another angle, we conjecture that if we exploit a set of simple motion analyzers to analyze each of the multiple motions, and then integrate their measurement results together based on the constraints among them, the performance may be much better than if we use just one big complicated motion analyzer. In other words, we expect that the "collaboration" among a set of simple motion analyzers will achieve better results for complex motion analysis. Such a *collaborative* approach indeed follows the methodology of *divide-conquer-combine*. There are two questions we are interested in answering:

- What characterizes the optimal integration of the measurements of the set of motion analyzers?
- Can we find an efficient computational diagram to obtain the optimal integration?

As a matter of fact, the concept of visual measurement integration is a bigger category. More specifically, we can integrate our prior knowledge into the visual estimator, or integrate the estimations from different visual cues such as edge and color [**130**], or integrate visual measurements from multiple scales [**41**, **73**], to obtain more robust solutions to computer vision problems. There are also several important issues of visual measurement integration that we need to address to achieve such a *collaborative* approach.

1.1.2. Visual measurement integration

There are many complications which affect the image formation process. Therefore, any visual measurement is doomed to have uncertainties. For example, to locate an object in the image scene, one classical technique is to match a shape model of the object with the image

gradient [**52**]. The matching process is inevitably smeared by the enormous edges presented in the background of the image. We must account for these uncertainties.

The complications in the image formation process may also make the visual measurement *multi-mode*, i.e., there may exist multiple results with high local confidences from that specific visual measurement, only one of them is the true solution though. We must handle the multi-modalities of the visual measurements well in the integration process to identify the true solution.

What is worse, the different visual measurement information may be inconsistent, i.e., two measurements may significantly deviate from each other but both are quite confident with themselves. The implication behind the inconsistency is that there must be false measurements. We must identify these false image measurements to exclude them from being integrated. Otherwise, these false visual measurements may completely degrade the integration results.

In summary, to achieve robust integration of multiple visual measurements, we must address the following three issues:

- How to model the uncertainties of visual measurement information?
- How to efficiently integrate different visual measurements which are *multi-mode*?
- How to integrate multiple visual measurements which may be inconsistent?

1.1.3. Distributed and parallel computing

There are two main causes necessitate the application of distributed and parallel computing in complex motion analysis: the heavy computation involved in the solution, and the *distributed information integration* resulted from the requirements of recovering each of the multiple correlated motions.

As we have mentioned, the high dimensional nature of complex motions potentially incurs exponential increase in computation demands to obtain a solution. This phenomenon is usually called the *curse-of-dimensionality*. To give a more concrete example, for articulated body tracking, under the joint angle representation, we have to seek the solution at least in a parametric space of 25 dimensions for the full human body. In a particle filtering [**52**, **53**] based top-down approach, the main computation involved is to evaluate the image observation of a sample hypothesis. However, the number of samples required to achieve a satisfactory solution is subject to an exponential increase with the dimensionality, so does the computational cost. The lack of scalability hinders the applicability of most existing methods for complex motion analysis.

One means to relieve the curse-of-dimensionality is to take a distributed representation, i.e., we can model each of the multiple correlated motions individually and reenforce the constraints at the same time. This is just like searching for the solution in a set of different but correlated manifolds, which compose the solution space. Based on distributed representations, efficient algorithms with close to linear complexity have been reported for articulated body tracking [**43**, **105**, **106**, **126**], human pose estimation [**27**, **47**, **94**, **95**], multiple object tracking [**137**] and structured deformable shape analysis [**45**, **46**],

Distributed representation advocates distributed information integration since the estimation of each of the multiple motions must be sought locally with the constraints from the others being reinforced. It is obvious that the solution to each component shares almost the same computational paradigm. This makes it possible to perform the computations involved in the solutions in a parallel fashion. Mostly, the solutions to the whole distributed model need to be performed iteratively to reach an equilibrium condition, which represents the final estimation of each of the multiple motions. Each iteration constitutes two steps:

- Compute the solution to each motion (can be in a parallel way) by combining the local estimation of the motion with the predictive estimations from the correlated motions through the constraints among them.
- Reformulate the constraints, and thus recalculate the neighborhood predictive estimations, based on the updated solution on each of the multiple correlated motions.

The ultimate solutions are the converged results from the iterations. It again reveals a *collaborative* solution to complex motion analysis.

1.1.4. Probabilistic reasoning and graphical models

Probabilistic theory provides a solid foundation for uncertainty reasoning and information integration. The uncertainty of a visual measurement can be conveniently captured by a probabilistic distribution, and probabilistic information integration [**16**, **72**] may be the most widely adopted integration approach. Under a probabilistic formulation, computer vision problems are usually formulated as the Bayesian inference of the hidden random variables, or the estimation of the model parameters, given the image data we observed.

Moreover, Bayesian theory conveniently enables us to incorporate any *a priori* knowledge or any reasonable assumptions about the solution of the problem of interest in the form of *prior distributions*. Meanwhile, the integration of different visual measurement information could be modeled as the interference of different random processes [**125**, **130**]. Nevertheless, we still need to address the same problems of information integration in Sec.1.1.2.

Graphical model [**58**] is a graph way of representing the factorizations of probabilistic systems. It can clearly and effectively present the correlations among the set of random variables in a principled way. There is a substantial literature [**26**, **29**, **31**, **41**, **43**, **46**, **47**, **51**, **110**, **111**, **120**, **125**, **126**, **132**] to exploit graphical models to solve computer vision problems.

In fact, the information integration in many computer vision problems can be formulated as and illustrated by the Bayesian inference or learning process on graphical models, where the problems of uncertainty, multi-modality, and inconsistency in the visual measurements may all be addressed in a principled way. Moreover, the category of Bayesian inference algorithms developed based on probabilistic variational methods are highly suited to distributed/parallel computing which have been demonstrated in [**41**, **43**–**47**, **126**]. Please refer to Chapter 2 for more discussions on probabilistic inference on graphical models.

1.2. Motivation

The discussions in Sec.1.1 motivate us to exploit variational Bayesian inference algorithms on graphical models to address the fundamental challenges of complex motion analysis. Our ultimate goal is a *collaborative approach* to complex motion analysis from video. Before we can achieve that, we need to achieve the following tasks:

- A probabilistic information integration framework which addresses *measurement uncertainty*, *measurement multi-modality*, and even *measurement inconsistency* in a principled way.
- An efficient computational paradigm which performs the information integration in a distributed and parallel way.
- A theoretic sound and practical methodology to effectively implement these integration algorithms, more specifically, probabilistic inference algorithms.

The fulfillment of the first task will enable us to achieve more robust integration of multiple measurements, and thus more robust results for complex motion analysis. The achievement of the second task ensures that our solution to complex motion analysis be scalable, i.e., the computational cost may not explode dramatically when the degrees of freedom of the complex

motion increases. Last but not least, in many situations, we encounter the intractability of closed-form implementation of a theoretic sound probabilistic inference algorithm, especially when the probabilistic distributions involved are highly multi-mode. We are also interested in addressing these implementation issues in this dissertation.

1.3. Organization

In this dissertation, we develop effective and efficient probabilistic reasoning algorithms based on graphical models to address the fundamental challenges in visual analysis of complex motions. The remainder of this dissertation is organized as follows:

- Chapter 2 presents a brief but comprehensive review of graphical models. Different types of graphical models are introduced and their equivalence is discussed. We also introduce different probabilistic inference algorithms on graphical models, in particular, probabilistic variational methods.
- Using articulated body motion as a specific example, Chapter 3 presents a distributed approach to complex motion analysis from video. The motion of the articulated body is modeled by a distributed Markov network. The Bayesian inference is performed by a novel mean field Monte Carlo algorithm (MFMC), which combines mean field variational method with particle filtering. It incorporates a set of interactive lower dimensional particle filters to obtain the solutions in a collaborative way. It also reveals an intrinsic distributed information integration framework.
- Targeting on the multi-modality of the motion posteriors, Chapter 4 proposes a novel variational maximum a posteriori (VMAP) algorithm. By combining a Gaussian mean field variational analysis with a deterministic annealing scheme, VMAP pursues the optimal MAP estimate of the probabilistic system defined on graphical

models. Rigorous mathematic proof regarding the convergence of the algorithm is presented.

- For automatic initialization of the collaborative motion analyzer, Chapter 5 proposes a novel data driven belief propagation algorithm (DDBP), which in principle combines importance sampling with belief propagation. It is an principled parallel framework to combine *top-down* reasoning with *bottom-up* reasoning. Experiments on estimating 2D human poses from single images obtain satisfactory results.
- To handle the situations when there are inconsistent measurements, in Chapter 6, we present our preliminary theoretical investigation about the characteristics of measurement inconsistency on graphical models. Two rigorous algebraic conditions are presented to determine the consistency and inconsistency of pairwise measurements. In addition, a more general criterion is presented. Based on the theoretical analysis, a new information integration method is proposed and leads to encouraging results when applied to the task of part based ensemble tracking.
- In Chapter 7, we summarize the dissertation in the views of complex motion analysis, probabilistic inference on graphical models, as well as information integration. Some meaningful future research directions are also discussed in the end.

1.4. Contributions

The original contributions of this work range from visual analysis of complex motions, probabilistic inference on graphical models, and robust information integration. For Bayesian inference on graphical models, we have the following technical contributions:

- A novel mean field Monte Carlo algorithm (MFMC), which reveals a set of collaborative particle filters for the Bayesian inference on any graphical models (e.g.,

Markov networks). It also reveals an efficient distributed approach to integrating multiple visual measurements.

- A novel variational maximum a posteriori algorithm (VMAP), which combines a Gaussian mean field variational analysis with a deterministic annealing scheme. This is a general Bayesian inference algorithm to approach to the global MAP estimate on any graphical models. We provide rigorous proof of an information theoretic theorem, which ensures the convergence of the VMAP algorithm.
- A novel data driven belief propagation algorithm, which combines top-down reasoning with bottom-up reasoning in a unified way.

For probabilistic information integration based on graphical models, our main contribution is that we reveal an intrinsic relationship between the fixe-point of a probabilistic integration system defined on graphical models and the consistency and inconsistency of the multiple measurements being modeled. In the case that the uncertainty of each of the measurements from the different sources can be captured by a Gaussian distribution, we provide two rigorous algebraic conditions to judge the consistency and inconsistency of pairwise measurements. A more general criterion is also presented to evaluate consistency and inconsistency in a general setting. All these criteria facilitate to exclude those false measurements from being integrated, and thus result in more robust visual inference results.

Most importantly, for complex motion analysis, our main contribution is a distributed and collaborative computational framework, which efficiently handles the fundamental challenges of visual analysis of complex motions. We stress beforehand here that although the main application we have discussed in this dissertation is on visual analysis of complex motion, the proposed approaches are all very general, which may also be applied to other computer vision problems.

CHAPTER 2

Probabilistic inference on graphical models: a brief review

2.1. Introduction

Probabilistic inference refers to the process of recovering some unknown random *factors* of a system from some observed data, all under the reasoning of probabilistic theory. It is an attractive approach to uncertainty reasoning and empirical learning [**86**] that has been widely used in computer vision, pattern recognition, artificial intelligence, bio-informatics, and economics, to list a few.

Denote $\mathcal{U}$ as the unknown *factors* of the system, e.g., it could be the unobservable *latent random variables*, the parameters of the probabilistic model that represents the system, and even the model itself. Also we denote Z as the observed data. Then, probabilistic inference algorithms intend to compute useful probabilistic quantities, e.g., the posterior probability $P(\mathcal{U}|Z)$, or to compute useful information theoretic quantities, e.g., the conditional entropy $H(\mathcal{U}|Z)$, or even compute the estimate of $\mathcal{U}$ based on some optimal criteria, e.g., the maximum a posterior estimation (MAP) $\hat{\mathcal{U}} = \arg\max_{\mathcal{U}} P(\mathcal{U}|Z)$ or the commonly used maximum likelihood estimation (ML) $\hat{\mathcal{U}} = \arg\max_{\mathcal{U}} P(Z|\mathcal{U})$ when $\mathcal{U}$ represents the probabilistic model or the model parameters.

Due to the flexibility of incorporating useful prior information into the probabilistic model, Bayesian (posterior) inference is favored. Although the MAP estimation has been criticized for being *atypical* and *basis dependent* [**78**], it prevents the over-fitting problem in the maximum likelihood estimation. Therefore, this chapter mainly discusses Bayesian

inference methods. In the mean time, we also use some spaces to discuss maximum likelihood methods, mainly in section 2.2.2. According to the genre of $\mathcal{U}$, the basic problems in probabilistic inference can be categorized into

(1) LATENT VARIABLE INFERENCE: the $\mathcal{U}$ represents the unobservable states of the system. We usually assume that the probabilistic model as well as the model parameters are known. While if either of the two is not specified, we might need to perform *parameter averaging* or *model averaging*.

(2) PARAMETER ESTIMATION: the $\mathcal{U}$ represents the model parameters. We can assume that the probabilistic model has been specified here.

(3) MODEL SELECTION: the $\mathcal{U}$ represents the probabilistic model which may be chosen from a set of different models . The task is to determine which model is the best one given the observed data Z.

However, as we may have noticed, the problem of *parameter estimation* is highly correlated to *model selection* in the sense that different parameters do result in different models. But the context of *model selection* is much larger because we may select among models of different types. While the parameter estimation could only be performed before the types of the models were specified. In addition, we may encounter problems in which we need to solve several basic problems listed above together. For example, we may need to jointly perform latent variable inference and parameter estimation, or model selection and parameter estimation.

Graphical model [**7**, **58**, **59**] is a powerful means of modeling multi-variate complex probabilistic systems. It represents the factorization of the joint distribution of a probabilistic system in a graph way, where mostly the nodes of the the graph denote the random variables,

and the edges of the graph, either directed or undirected, denote the probabilistic quantities defined on the set of connected random variables represented by the nodes. As we will also discuss in Sec. 2.3, there are also graphical models in which there are two types of nodes, namely variable nodes and function nodes, and the edges are not associated with any probabilistic quantities [**68**]. The function nodes explicitly model the probabilistic correlations among the set of variable nodes connected to them.

Because of the convenience in visualizing the correlations among the set of random variables of the probabilistic system, probabilistic reasoning with graphical models become a popular approach, especially in the literature of computer vision [**28**,**29**,**31**,**41**,**51**,**105**,**110**, **125**,**132**]. In this chapter, we present a introduction review of probabilistic reasoning with graphical models. The rest of the chapter is organized as follows: in Sec. 2.2, we present the general formulation of each of the basic problems of probabilistic inference. Then in Sec. 2.3, we introduce various types of graphical models for probabilistic reasoning. After that, we present a review of different probabilistic inference methods in Sec. 2.4. Finally, we conclude this chapter with some discussion remarks in Sec. 2.5.

2.2. Basic problems of probabilistic inference

Before continuing the discussions of the formulation of each of the basic problems, we firstly present the notations that we use throughout this chapter in Table 2.1. In summary, we adopt bold-face letter to represent a random variable and normal-face letter to represent a specific value that a random variable takes. In addition, a probabilistic system can usually be represented or be defined by a joint distribution, i.e.,

$$\mathbf{ps} \sim P(\mathbf{Z}, \mathbf{X}, \mathbf{\Theta}, \mathcal{H}). \tag{2.1}$$

$\mathbf{ps}$	A probabilistic system
$\mathbf{z}$	Observed data or observable random variable
$\mathbf{Z}$	$\{\mathbf{z}_1, \ldots, \mathbf{z}_n\}$
$\mathbf{x}$	Latent unobservable random variable
$\mathbf{X}$	$\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$
$\mathcal{H}$	The probabilistic model
$\mathbf{\Theta}$	The set of parameters of $\mathcal{H}$
$\mathcal{U}$	Unknown factors of a probabilistic system, which could be any of $\mathbf{x}$, $\mathbf{\Theta}$ and $\mathcal{H}$ or a subset of them
z	A concrete value of $\mathbf{z}$
Z	$\{z_1, \ldots, z_n\}$
x	A concrete value of $\mathbf{x}$
X	$\{x_1, \ldots, x_n\}$
θ	A concrete parameter set of the probabilistic model
h	A specific type of probabilistic model

Table 2.1. Detailed notations.

2.2.1. Latent variable inference

Assume that both the probabilistic model $\mathcal{H}$ and the model parameters $\mathbf{\Theta}$ be known, then the joint probability of the system becomes

$$\mathbf{ps} \sim P(\mathbf{Z}, \mathbf{X}) \sim P(\mathbf{Z}, \mathbf{X}|\theta, h). \tag{2.2}$$

Suppose we observe that $\mathbf{Z} = Z$. Then, the inference is to recover the posterior density

$$P(\mathbf{X}|\mathbf{Z} = Z). \tag{2.3}$$

According to the Bayesian rule, we have

$$P(\mathbf{X}|\mathbf{Z} = Z) = \frac{P(\mathbf{Z} = Z, \mathbf{X})}{P(\mathbf{Z} = Z)}, \tag{2.4}$$

where

$$P(\mathbf{Z} = Z) = \int_{\mathbf{X}} P(\mathbf{Z} = Z, \mathbf{X}) d\mathbf{X} \tag{2.5}$$

is also called partition function or evidence. It is the general difficulty in evaluating partition functions that makes the Bayesian inference not trivial to achieve.

2.2.2. Parameter estimation

Assume that the probabilistic model $\mathcal{H}$ be known and all the system random states could be observed, but the model parameters $\mathbf{\Theta}$ are unknown, then we immediately have the likelihood probability, i.e.,

$$P(\mathbf{Z}|\mathbf{\Theta}, h). \tag{2.6}$$

A common technique for estimating $\mathbf{\Theta}$ is the *maximum likelihood* (ML) estimation. Suppose $\mathbf{Z} = \{z_1, z_2, \ldots, z_N\}$ contains N *i.i.d.* samples from $P(\mathbf{Z}|\mathbf{\Theta}, h)$, then the likelihood of observing these samples is

$$\mathcal{L}(\mathbf{\Theta}) = \prod_i P(z_i|\mathbf{\Theta}, h). \tag{2.7}$$

The ML estimation $\hat{\theta}_{ML}$ is such that

$$\hat{\theta}_{ML} = \arg\max_{\mathbf{\Theta}} \mathcal{L}(\mathbf{\Theta}). \tag{2.8}$$

If we have a priori knowledge $P(\mathbf{\Theta})$ (without affecting the clarity, we will not show h to simplify the notation), then we can also apply the Bayesian rule to achieve the MAP estimation, i.e.,

$$\begin{cases} P(\mathbf{\Theta}|\mathbf{Z}) \propto P(\mathbf{Z}|\mathbf{\Theta})P(\mathbf{\Theta}) \\ \hat{\theta}_{MAP} = \arg\max_{\mathbf{\Theta}} P(\mathbf{Z}|\mathbf{\Theta})P(\mathbf{\Theta}) \end{cases}. \tag{2.9}$$

While a more general problem is that there are also unobservable random states $\mathbf{X}$ in the system, then the ML estimation of $\mathbf{\Theta}$ can be obtained by the well informed EM algorithm [**21**] for maximum likelihood estimation from incomplete data. The essence of the EM algorithm is encoded in the following equation set. It basically involves the iteration of two steps, i.e., the E-Step and the M-Step,

$$\begin{cases} & \mathcal{L}(\mathbf{\Theta}) = P(\mathbf{X}, \mathbf{Z}|\mathbf{\Theta}) \\ E-Step: & Q(\mathbf{\Theta}, \mathbf{\Theta}^*) = E\left[\log P(\mathbf{X}, \mathbf{Z}|\mathbf{\Theta})|\mathbf{Z}, \mathbf{\Theta}^*\right] \\ M-Step: & \mathbf{\Theta}^* = \arg\max_{\mathbf{\Theta}} Q(\mathbf{\Theta}, \mathbf{\Theta}^*) \end{cases} . \tag{2.10}$$

where both steps guarantee to increase $\mathcal{L}(\mathbf{\Theta})$, and $\mathbf{\Theta}^*$ represents the estimation of $\mathbf{\Theta}$ in the previous M-Step.

What is more, the idea behind the maximum likelihood EM algorithm can also be applied to Bayesian estimation of the model parameters with incomplete data by deriving steps similar to Eq. 2.10. This refers to the Bayesian variational EM algorithm [**7**] for estimating model parameters, which can also be used for model selection.

2.2.3. Model selection and model averaging

Before proceeding to the discussion of model selection, we will firstly introduce *Occam's Razor*, which is an automatical principle (or the *principle of parsimony*) in all scientific modeling. It is stated as follows, *"one should not increase, beyond what is necessary, the number of entities required to explain anything."*

The problem of Bayesian model selection is as follows: suppose the probabilistic model $\mathcal{H}$ could be from a set of models $\mathbf{S}_{\mathcal{H}} = \{h_1, h_2, \ldots, h_N\}$, then based on the observed data $\mathbf{Z}$,

the MAP selection of the model is

$$\hat{\mathbf{h}}_{MAP} = \arg\max_{\mathcal{H}} P(\mathcal{H}|\mathbf{Z}) \tag{2.11}$$

Actually, this tells us how to perform the model comparison, i.e.,

$$\frac{P(h_1|\mathbf{Z})}{P(h_2|\mathbf{Z})} = \frac{P(\mathbf{Z}|h_1)}{P(\mathbf{Z}|h_2)}\frac{P(h_1)}{P(h_2)} \tag{2.12}$$

The first factor on the right side of Eq. 2.12 is called the *Bayesian factor* or *likelihood ratio*.

Suppose there are also unknown parameters $\mathbf{\Theta}_i$ related to each model h_i, then the likelihood terms are usually obtained by integrating the parameters out, e.g.,

$$P(\mathbf{Z}|h_i) = \int_{\mathbf{\Theta}_i} P(\mathbf{Z}|\mathbf{\Theta}_i, h_i)P(\mathbf{\Theta}_i|h_i)d\mathbf{\Theta}_i, i = 1, \ldots, N \tag{2.13}$$

where $P(\mathbf{\Theta}_i|h_i)$ is the prior probability of the model parameters.

Then the probabilistic density at a new data Z_1, which is also called *predictive probability*, could be obtained by averaging over all the models and all the parameters, i.e.,

$$P(Z_1|\mathbf{Z}) = \sum_{i=1}^{N} P(h_i|\mathbf{Z}) \int_{\mathbf{\Theta}_i} P(Z_1|\mathbf{\Theta}_i, h_i)P(\mathbf{\Theta}_i|h_i)d\mathbf{\Theta}_i. \tag{2.14}$$

This is also called *model averaging* as it accounts for all the uncertainties on both models and parameters. It may not be that straightforward, but the Bayesian factor embodies Occam's Razor automatically as it will favor models with simple form (i.e., with less free parameters [**78**]) since more complex models can *a priori* model a larger range of data set and thus distribute the probability more.

2.3. Graphical model representation

Just as we have mentioned, a probabilistic system is usually defined by its joint probability. In most real applications, the joint probability may be factorized due to the conditional independencies, or local only correlations among the set of random variables. That is the reason why graphical models have been favored in representing probabilistic systems because they can neatly present and visualize such kind of conditional independencies or local correlations. Then, under the graphical model representation, the three basic problems of probabilistic inference become *latent variable inference*, *parameter learning* and *structure learning*, respectively. In this section, we will briefly review different types of graphical models which are widely used in the literature. Actually, these graphical models can be transformed into one another through certain topological and mathematical manipulations. In this sense, they are equivalent [**32**, **58**].

2.3.1. Belief networks, Bayesian network, and dynamic Bayesian network

Bayesian network or belief network is a way of presenting a particular joint distribution factorizations based on directed graphical models. For example, let $\mathbf{X}_{\mathcal{S}} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_6\}$ and the joint probability be

$$P(\mathbf{X}_{\mathcal{S}}) = P(\mathbf{x}_1)P(\mathbf{x}_2)P(\mathbf{x}_4|\mathbf{x}_1)P(\mathbf{x}_5|\mathbf{x}_2)P(\mathbf{x}_3|\mathbf{x}_1, \mathbf{x}_2)P(\mathbf{x}_6|\mathbf{x}_4, \mathbf{x}_5). \tag{2.15}$$

The Bayesian network representing such a joint distribution factorization is shown in Fig. 2.1. The arrows in the Bayesian network shown in Fig. 2.1 are corresponding to the conditional probabilities in Eq. 2.15. The node in the start of the arrow is called the parent of the node in the end of the arrow. Actually, the general factorization of a Bayesian network is as

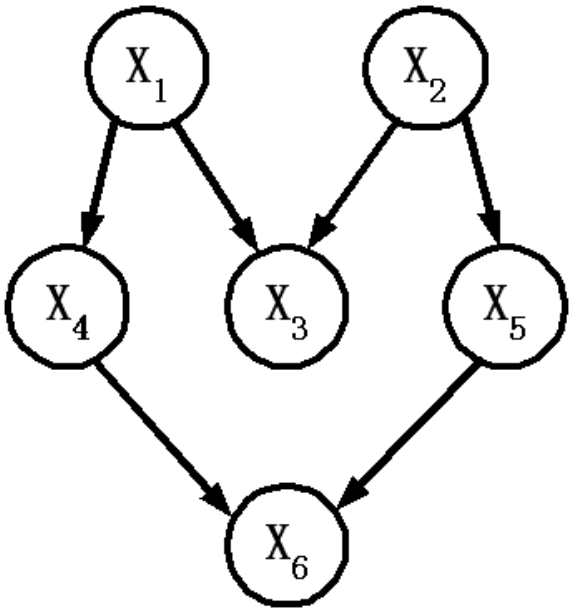


Figure 2.1. A Bayesian network.

follows, suppose $\mathbf{X}_{\mathcal{S}} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, then

$$P(\mathbf{X}_{\mathcal{S}}) = \prod_{i=1}^{N} P(\mathbf{x}_i | Par(\mathbf{x}_i)), \tag{2.16}$$

where $Par(\mathbf{x}_i)$ denotes the set of all the parent nodes of $\mathbf{x}_i$.

A Bayesian network is said to be singly connected if there is no undirected loop in it. For example, the Bayesian network in Fig. 2.1 is not singly connected. While *dynamic Bayesian network is a special set of singly connected Bayesian networks that aim at time series modeling* [**85**]. The hidden Markov model (HMM) [**91**] may be one of the most widely used dynamic Bayesian network.

2.3.2. Markov random field and Markov network

The Markov random field (MRF) [**71**] or *Markov network* is another means of representing joint distribution factorizations based on indirected graphical models. It is naturally suited

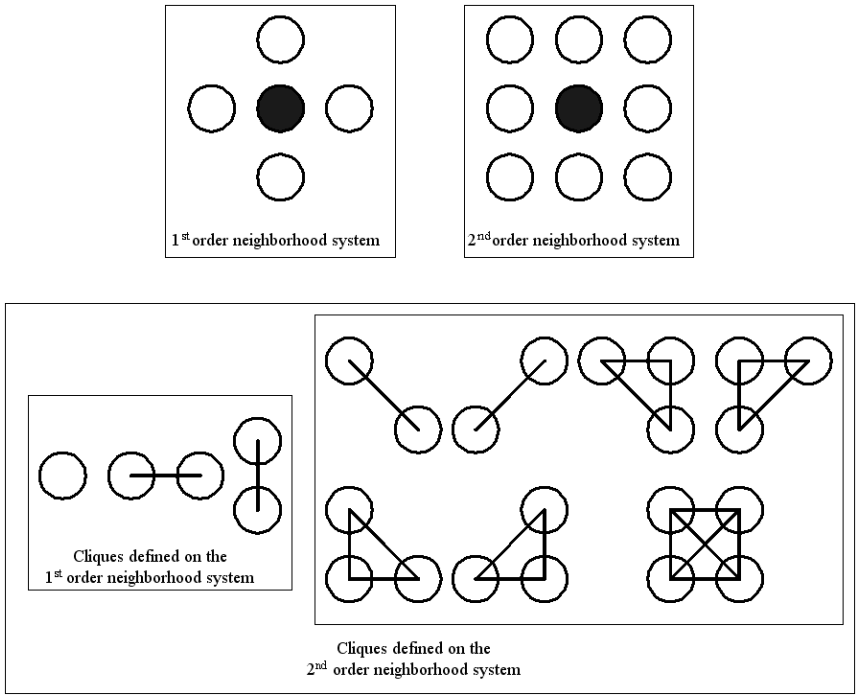


Figure 2.2. Neighborhood system and cliques.

to model images. Basically, for modeling images, it involves choosing a neighborhood system in the image coordinate system $\mathbf{\Omega}$, then the substructure in the neighborhood system with fully connected nodes are called *cliques*. We can then define the potential $\mathcal{V}_{\mathcal{C}}(\mathbf{X}_{\mathcal{C}})$ on the cliques of the neighborhood system, while $\mathbf{X}_{\mathcal{C}}$ is the set of random variables representing pixel values at the image locations in the clique.

Assume the set of all pixel values of image $\mathcal{I}$ be $\omega = \{x_{ij}, \{i,j\} \in \mathbf{\Omega}\}$, then the probability of seeing image $\mathcal{I}$ is a *Gibbs distribution*, i.e.,

$$P(\omega) = \frac{1}{C_{\mathbf{\Omega}}} \exp\left(-\sum_{\mathcal{C}} \mathcal{V}_{\mathcal{C}}(\mathbf{X}_{\mathcal{C}})\right) \tag{2.17}$$

where

$$C_{\mathbf{\Omega}} = \int_{\mathbf{\Omega}} \exp\left(-\sum_{\mathcal{C}} \mathcal{V}_{\mathcal{C}}(\mathbf{X}_{\mathcal{C}})\right) d\omega \tag{2.18}$$

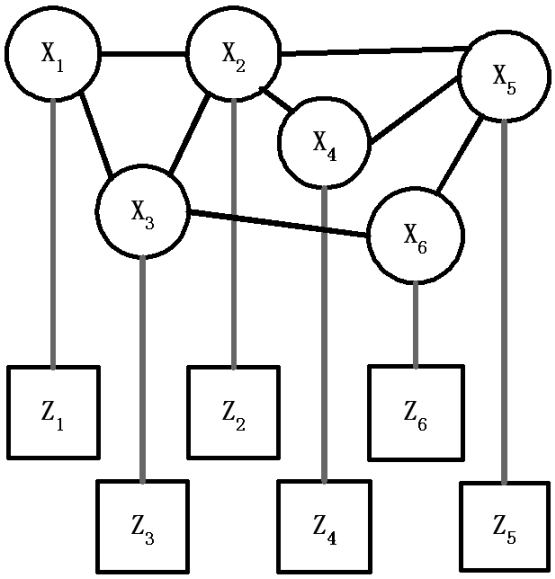


Figure 2.3. A Markov network.

Fig. 2.2 shows some typical neighborhood systems and the associated cliques on them (for other higher order cliques, please refer to [**33**]). Indeed, the equivalence between a *Markov random field* and a *Gibbs random field* defined by a *Gibbs distribution* has been proven in [**33**].

There is a special type of widely applied MRF where only pairwise potentials are used. An example of such kind of Markov networks is shown in Fig. 2.3. The joint probability of a Markov network is defined as

$$P(\mathbf{X}, \mathbf{Z}) = \frac{1}{C_Q} \prod_{\{i,j\} \in \mathcal{E}} \psi_{ij}(\mathbf{x}_i, \mathbf{x}_j) \prod_{i \in \mathcal{O}} \phi_i(\mathbf{x}_i, \mathbf{z}_i), \tag{2.19}$$

where C_Q is a normalization constant, $\mathcal{E}$ denotes the set of links among the latent variables, $\mathcal{O}$ denotes the set of links between a latent variable and an observed variable, and both $\psi_{ij}(\mathbf{x}_i, \mathbf{x}_j)$ and $\phi_i(\mathbf{x}_i, \mathbf{z}_i)$ are called *potential functions* (in fact, exponential of negative potentials). We can clearly notice that Markov networks are purely undirected graphical models.

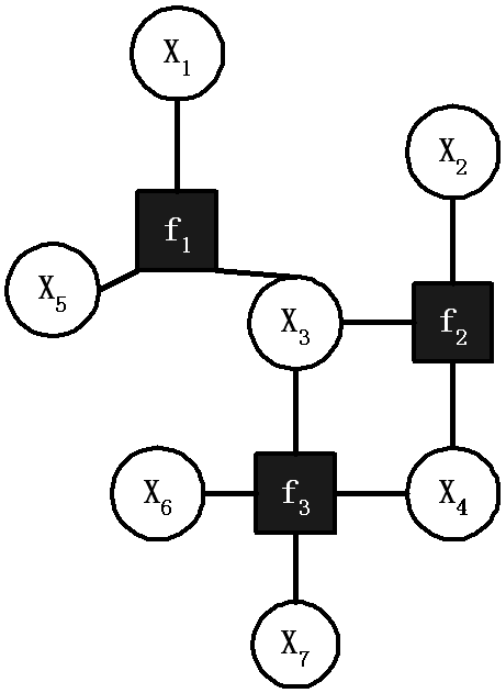


Figure 2.4. A factor graph.

2.3.3. Factor graph

While the *conditional probabilities* and the *potential functions* are associated with directed edges on Bayesian networks and undirected edges on Markov networks, respectively, the factor graph [**68**] explicitly defines two types of nodes on a bipartite graph, i.e., the *variable node* and the *function node*.

Fig. 2.4 presents an example of a factor graph with 7 variable nodes (white circle) and function nodes (black box), each function node defines a function of all the variable nodes connected to it. Let $\mathbf{X}_{\mathcal{S}} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_7\}$, the joint probability defined in this factor graph is

$$P(\mathbf{X}_{\mathcal{S}}) = \frac{1}{C_f} f_1(\mathbf{x}_1, \mathbf{x}_3, \mathbf{x}_5) f_2(\mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4) f_3(\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_6, \mathbf{x}_7), \tag{2.20}$$

where C_f is again a normalization constant. More generally, for a factor graph with N function nodes, the joint probability is

$$P(\mathbf{X}_{\mathcal{S}}) = \frac{1}{C_f} \prod_{i=1}^{N} f_i(\mathbf{X}_{f_i}), \tag{2.21}$$

where $\mathbf{X}_{f_i}$ represents the set of variables attached to the function node f_i.

2.3.4. Equivalence of the different graphical models

As a matter of fact, these three types of graphical models can topologically and mathematically be transformed to one another, i.e., they represent the same distribution factorizations. For example, a directed Bayesian network may be converted to an undirected high-order Markov network through the process called *moralization*. A broad discussion of the transformations among different graphical models can be found in [**32**, **58**].

2.4. Probabilistic inference algorithm

For most real applications, we encounter situations where we must perform joint latent variable inference and parameter estimation, or joint latent variable inference and model selection. For both tasks, the latent variable inference is essential for the solution. Therefore, in this section, we mainly discuss algorithms for latent variable inference. We also discuss some methods for model selection and parameter learning in the last part of this section.

2.4.1. Exact inference algorithm

2.4.1.1. Sum-product algorithm and belief propagation. If there is no loop in the graphical model (i.e., tree structured). There can be efficient propagation algorithms to

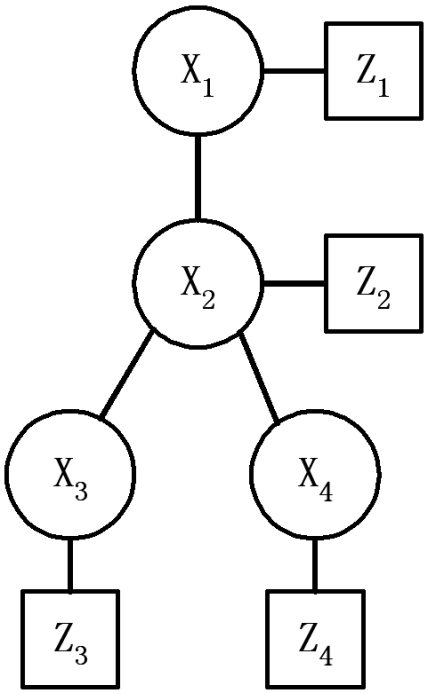


Figure 2.5. Message passing example.

perform the *exact* inference of the marginal posterior distributions or maximum marginal estimate, which is called the belief propagation algorithm (i.e., sum-product or max-product) [**29**, **58**,**82**,**134**]. In fact, the famous forward-backward algorithms [**85**,**91**] and Viterbi algorithm for hidden Markov model (HMM) are special cases of sum-product and max-product, respectively. Not losing any generality, our discussions are presented under Markov networks.

For tree structured Markov networks, the sum-product belief propagation [**28**,**29**] can be expressed by the following two equations:

$$\mathbf{m}_{ij}(\mathbf{x}_i) = \int_{\mathbf{x}_j} \phi_j(\mathbf{x}_j, \mathbf{z}_j)\psi_{ij}(\mathbf{x}_j, \mathbf{x}_i) \prod_{k\in\{\mathcal{N}(j)\setminus i\}} \mathbf{m}_{jk}(\mathbf{x}_j)d\mathbf{x}_j \tag{2.22}$$

$$P(\mathbf{x}_i|\mathbf{Z}) = \frac{1}{\mathbf{Z}_{\mathcal{Q}}}\phi_i(\mathbf{x}_i, \mathbf{z}_i) \prod_{k\in\mathcal{N}(i)} \mathbf{m}_{ik}(\mathbf{x}_i). \tag{2.23}$$

The running of the belief propagation algorithm involves the iteration of Eq. 2.22 to conver-

gence. The number of iterations necessary in the tree structured case equals to the *depth* of the graphical model, i.e., the length of the longest path in the graphical model. We will use a simple example to demonstrate the correctness of the algorithm. Suppose a probabilistic distribution factorization is represented as the graphical model shown in Fig. 2.5, and we would want to evaluate $P(\mathbf{x}_1|\mathbf{Z})$, the calculation is as follows

$$\begin{aligned} P(\mathbf{x}_1|\mathbf{Z}) &= \frac{1}{C}\int_{\mathbf{x}_2}\int_{\mathbf{x}_3}\int_{\mathbf{x}_4}\phi(\mathbf{x}_1,\mathbf{z}_1)\phi(\mathbf{x}_2,\mathbf{z}_2)\phi(\mathbf{x}_3,\mathbf{z}_3)\phi(\mathbf{x}_4,\mathbf{z}_4)\psi(\mathbf{x}_1,\mathbf{x}_2)\psi(\mathbf{x}_2,\mathbf{x}_3)\psi(\mathbf{x}_2,\mathbf{x}_4) \\ &= \frac{1}{C}\int_{\mathbf{x}_2}\phi(\mathbf{x}_1,\mathbf{z}_1)\phi(\mathbf{x}_2,\mathbf{z}_2)\psi(\mathbf{x}_1,\mathbf{x}_2)\underbrace{\int_{\mathbf{x}_3}\phi(\mathbf{x}_3,\mathbf{z}_3)\psi(\mathbf{x}_2,\mathbf{x}_3)}_{\mathbf{m}_{23}(\mathbf{x}_2)}\underbrace{\int_{\mathbf{x}_4}\phi(\mathbf{x}_4,\mathbf{z}_4)\psi(\mathbf{x}_2,\mathbf{x}_4)}_{\mathbf{m}_{24}(\mathbf{x}_2)} \\ &= \frac{1}{C}\phi(\mathbf{x}_1,\mathbf{z}_1)\underbrace{\int_{\mathbf{x}_2}\phi(\mathbf{x}_2,\mathbf{z}_2)\psi(\mathbf{x}_1,\mathbf{x}_2)\mathbf{m}_{23}(\mathbf{x}_2)\mathbf{m}_{24}(\mathbf{x}_2)}_{\mathbf{m}_{12}(\mathbf{x}_1)} \\ &= \frac{1}{C}\phi(\mathbf{x}_1,\mathbf{z}_1)\mathbf{m}_{12}(\mathbf{x}_1), \end{aligned} \tag{2.24}$$

where C is the normalization constant. It is easy to figure out that Eq. 2.24 is a specific example of Eq. 2.23.

2.4.1.2. Junction tree algorithm. When there are loops in the graphical model, direct applying the belief propagation algorithm will not achieve the exact inference results. However, there exists more general algorithm, namely junction tree algorithm, which can perform exact inference by the message passing process among clusters of nodes in loopy graphical models. The junction tree algorithm usually involves the following steps.

(1) If the graphical model is directed, moralize it to be an undirected graphical model.

(2) Determining an elimination order of the nodes in the graphical model, i.e., triangulate the undirected graph.

(3) Constructing the junction tree from the elimination order. This involves clustering the nodes as several sets called *super node* and weight the edges among different super nodes by the *separator node* set.

(4) Message passing among super nodes and separator nodes in the junction tree to perform the inference.

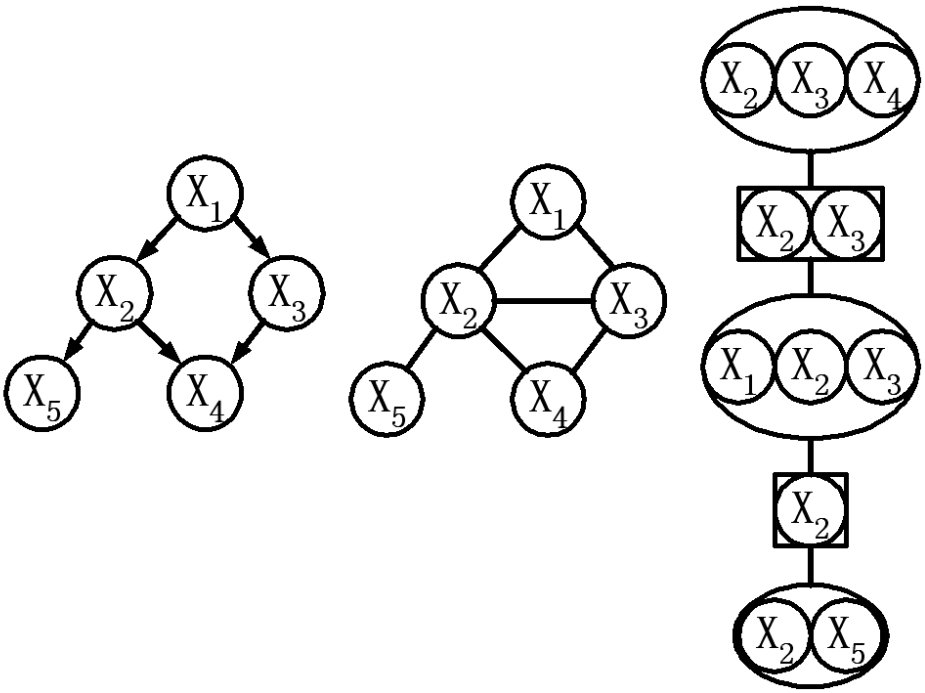
In the junction tree, assume each *super node* $\mathbf{X}_{sn}$ has the potential $\psi(\mathbf{X}_{sn})$ and each *separator node* $\mathbf{X}_{sep}$ has the evidence $\phi(\mathbf{X}_{sep})$. Then the message propagation between two *super-node* $\mathbf{X}_{sn}^{\mathbf{V}}$ and $\mathbf{X}_{sn}^{\mathbf{W}}$, which are separated by the *separator node* $\mathbf{X}_{sep}^{\mathbf{S}}$ ($\mathbf{V}$, $\mathbf{W}$ and $\mathbf{S}$ denote the set of normal nodes in the super node or the separator node), are evaluated according to the following two equations

$$\begin{cases} \phi^*(\mathbf{X}_{sep}^{\mathbf{S}}) = \int_{\mathbf{X}_{sn}^{\mathbf{V}} \setminus \mathbf{X}_{sep}^{\mathbf{S}}} \psi(\mathbf{X}_{sn}^{\mathbf{V}}) \\ \psi^*(\mathbf{X}_{sn}^{\mathbf{W}}) = \frac{\phi^*(\mathbf{X}_{sep}^{S})}{\phi(\mathbf{X}_{sep}^{S})} \psi(\mathbf{X}_{sn}^{\mathbf{W}}) \end{cases} . \tag{2.25}$$

Example 2.4.1. *Fig. 2.6 shows an example of applying the junction tree algorithm on a Bayesian network, which follows exactly the same procedures as presented above.*

2.4.2. The variational approach

As we may have noticed, exact inferences on complex probabilistic systems are not always feasible to achieve. Variational approach provides a principled way for approximate inference. Instead of trying to recover the posterior distribution directly, it usually involves the construction of a variational distribution. Then a lower-bound of the *data likelihood* will be maximized by minimizing the KL divergence between the variational distribution and the true posterior distribution. The key of the theoretic deduction of variational inference



(a) Original BN (b) Moralized MN (c) Junction tree

Figure 2.6. Example of junction tree. In (c), triangles denote separator nodes and ellipses denote super nodes.

methods is based on the Jensen's inequality, i.e.,

$$\ln P(\mathbf{Z}) = \ln \int_{\mathcal{U}} P(\mathbf{Z},\mathcal{U}) d\mathcal{U} = \ln \int_{\mathcal{U}} Q(\mathcal{U}) \frac{P(\mathbf{Z},\mathcal{U})}{Q(\mathcal{U})} d\mathcal{U} \geq \int_{\mathcal{U}} Q(\mathcal{U}) \ln \frac{P(\mathbf{Z},\mathcal{U})}{Q(\mathcal{U})} d\mathcal{U}. \quad (2.26)$$

Maximizing the lower bound with respect to the variational distribution $Q(\mathcal{U})$ (actually, it is achieved when the equality holds) will result in

$$Q(\mathcal{U}) = P(\mathcal{U}|\mathbf{Z}), \quad (2.27)$$

which is exactly the posterior distribution we would want to estimate. However, exact optimization of $Q(\mathcal{U})$ to achieve the maximum of the lower-bound is also intractable. In practice, *one always constrains the distribution $Q(\mathcal{U})$ to make the optimization more mathematically*

tractable while also ensures the approximation to be as accurate as possible [**58**, **59**]. We discuss some of the widely used variational distributions in the following sections.

2.4.2.1. Mean field variational method. The most widely used variational distribution $Q(\mathcal{U})$ might be the ones that have the fully factorized form. More specifically, assume $\mathcal{U} = \{\mathcal{U}_1, \ldots, \mathcal{U}_\mathcal{L}\}$, we let

$$Q(\mathcal{U}) = \prod_{i=1}^{\mathcal{L}} Q_i(\mathcal{U}_i), \tag{2.28}$$

which is also called the mean field variation. Substituting Eq. 2.28 into Eq. 2.26, with the constraints that

$$\int_{\mathcal{U}_i} Q_i(\mathcal{U}_i) d\mathcal{U}_i = 1, \quad i = 1, \ldots, \mathcal{L} \tag{2.29}$$

we can formulate a Lagrangian multipliers from the lower bound and apply the elementary calculus of variations by taking the functional variation of the Lagrangian w.r.t. each of the factorized distribution $Q_i(\mathcal{U}_i)$, we can easily obtain the following set of fixed-point equations, i.e.,

$$Q_i(\mathcal{U}_i) = \frac{1}{C_i} \exp\left\{ \int_{\mathcal{U} \setminus \mathcal{U}_i} \prod_{j=1 \ldots \mathcal{L}}^{j \neq i} Q_j(\mathcal{U}_j) \log P(\mathbf{Z}, \mathcal{U}) d\mathcal{U} \right\}, \quad i = 1, \ldots, \mathcal{L} \tag{2.30}$$

where the C_i is the normalization constant. It is easy to figure out that when $\mathcal{U}$ contains both the latent variables $\mathbf{X}$ and model parameters $\mathbf{\Theta}$, the factorization of these two will result in the variational Bayesian EM algorithm [**7**], which can be deemed as a generalization of the traditional EM algorithm [**21**].

2.4.2.2. Structured variational method. Just as the junction tree algorithm exploits the local substructure of a graphical model to perform the message passing, the *structured variational method* utilizes the local substructures of a graphical model to achieve more accurate as well as more tractable variational approximation for the probabilistic inference.

There are actually only two principles for choosing the structured variational distributions, i.e.,

(1) The structured variational distributions must be mathematically more tractable.
(2) The structure of the graphical model representing the variational distribution should be as close as possible to that of the original graphical model.

With the structured variational factorization, among different substructures, we perform the mean field variational analysis. While inside each of the substructure, we can then perform the exact inference using a suitable exact inference algorithm. A more detailed discussion can be found in [**54**].

2.4.2.3. The free energy understanding of BP and mean field method. As we have discussed, when there are loops in the graphical model, the local message passing belief propagation is not able to obtain the exact inference results. But many empirical studies show that if we still perform the belief propagation on loopy graphical models, we may still recover satisfactory approximate inference results. This motivates many researchers to study the theoretical explanation of this phenomenon [**134**, **135**]. Indeed, there are both physical free energy explanation and mathematical explanation based on the calculus of variations. As a matter of fact, there is an unified understanding of the belief propagation algorithm and the mean field variational method, all based on the variational free energy approximation. More detailed discussions can be found in [**134**]. Here we present a brief summary of the content presented in [**134**] with our own notation and presentation.

Denote $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{\mathcal{L}}\}$ as the set of all the random variables, and assume the probabilistic system defined on $\mathbf{X}$ can be represented by a *high-order Markov network*, then

the joint probability is defined as

$$P(\mathbf{X}) = \frac{1}{C_{\mathcal{Q}}} \prod_{\mathcal{C}} \psi_{\mathcal{C}}(\mathbf{X}_{\mathcal{C}}) \tag{2.31}$$

$$= \frac{1}{C_{\mathcal{Q}}} \exp\left(- \sum_{\mathcal{C}} \mathcal{V}_{\mathcal{C}}(\mathbf{X}_{\mathcal{C}}) \right) \tag{2.32}$$

$$= \frac{1}{C_{\mathcal{Q}}} \exp\left(-E(\mathbf{X})\right) \tag{2.33}$$

Bayesian inference needs to calculate the *Helmholtz* free energy F_H, which is

$$F_H = -\ln C_{\mathcal{Q}}. \tag{2.34}$$

It is usually very difficult to evaluate the *Helmholtz* free energy since it may involve multiple integrals of the random variables with complex distributions. An important method to solve it is based on a variational approach. Suppose we have a variational distribution $B(\mathbf{X})$ to approximate $P(\mathbf{X})$, then the *variational* free energy or the *Gibbs* free energy is

$$F(B) = \int_{\mathbf{X}} B(\mathbf{X}) E(\mathbf{X}) \mathbf{dx} + \int_{\mathbf{X}} \mathbf{B}(\mathbf{X}) \ln \mathbf{B}(\mathbf{X}) \mathbf{dX} \tag{2.35}$$

$$= U(B) - H(B) \tag{2.36}$$

$$= F_H + KL(B(\mathbf{X}) \| P(\mathbf{X})) \tag{2.37}$$

where $U(B)$ is called the *variational average energy*, $H(B)$ is called the *variational entropy*, and $KL(\cdot)$ is the KL divergence between the two distributions.

Again, it is easy to figure out that we can minimize the KL divergence to zero to achieve the exact results, but that is still intractable. Usually we can only minimize it on a specific

set of distributions. For example, if we let

$$B(\mathbf{X}) = \prod_i B_i(\mathbf{x}_i), \tag{2.38}$$

we can then approximate the *Helmholtz* free energy by the so-called *mean field* free energy. This indeed results in the mean field variational method.

Another means of free energy approximation is the so-called *Bethe* free energy. This includes the approximation as

$$B(\mathbf{X}) = \frac{\prod_{\mathcal{C}} B_{\mathcal{C}}(\mathbf{X}_{\mathcal{C}})}{\prod_{i=1}^{\mathcal{L}} (B_i(\mathbf{x}_i))^{d_i - 1}} \tag{2.39}$$

where d_i is the number of cliques that the node $\mathbf{x}_i$ belongs to. We also have the constraints that for any $\mathbf{x}_i \in \mathbf{X}_{\mathcal{C}}$,

$$B_i(\mathbf{x}_i) = \int_{\mathbf{X}_{\mathcal{C}} \setminus \mathbf{x}_i} B_{\mathcal{C}}(\mathbf{X}_{\mathcal{C}}). \tag{2.40}$$

We then have the Bethe approximation to the Gibbs free energy, i.e.,

$$F_{Bethe} = U_{Bethe} - H_{Bethe}, \tag{2.41}$$

where

$$U_{Bethe} = \sum_{\mathcal{C}} \int_{\mathbf{X}_{\mathcal{C}}} B_{\mathcal{C}}(\mathbf{X}_{\mathcal{C}}) \mathcal{V}_{\mathcal{C}}(\mathbf{X}_{\mathcal{C}}) d\mathbf{X}_{\mathcal{C}} \tag{2.42}$$

and

$$H_{Bethe} = -\sum_{\mathcal{C}} \int_{\mathbf{X}_{\mathcal{C}}} B_{\mathcal{C}}(\mathbf{X}_{\mathcal{C}}) \ln B_{\mathcal{C}}(\mathbf{X}_{\mathcal{C}}) d\mathbf{X}_{\mathcal{C}} + \sum_{i=1}^{\mathcal{L}} (d_i - 1) \int_{\mathbf{x}_i} B_i(\mathbf{x}_i) \ln B_i(\mathbf{x}_i) d\mathbf{x}_i, \tag{2.43}$$

Please note that the *Bethe* free energy could be exact when there is no loops in the graphical models. We can also figure out that the beliefs in the belief propagation algorithms are

the fixed-points of the *Bethe* free energy, subject to the constraints that all the beliefs are normalized and consistent [**134**]. Actually, any valid variational distributions will result in a free energy approximation, this is the foundation for the generalized belief propagation algorithm [**134**, **136**].

2.4.3. Monte Carlo methods

The Monte Carlo method is another type of approximate probabilistic inference algorithms. It is based on the strong law of large numbers, i.e., when the number of *i.i.d.* samples from a certain probabilistic distribution is large enough, then any order of the sample quadratures will converge, with probability one, to the same order of distribution statistics. The Monte Carlo algorithms are also called sampling algorithms. In this section, we will start from some basic sampling algorithms and then move on to more complex sampling methods.

It is generally difficult to draw samples directly from a complex probabilistic distribution, but it is assumed that we can conveniently evaluate the probability at one specific point. A discussion on why this is true could be found in [**78**].

2.4.3.1. Basic sampling methods.

Uniform sampling. A direct thought of sampling would be the uniform sampling technique. The idea is to uniformly sample the state space of $\mathbf{x}$ and evaluate $P(\mathbf{x})$, from which we would want to sample, at each sample point. However, uniform sampling is not enough, especially for probabilistic distributions in a high dimensional state space. The number of samples would be fairly huge to obtain satisfactory results. A discussion of this issue on medium size Ising model could also be found in [**78**].

Reject sampling. Again, suppose that $P(\mathbf{x})$ is too complicated to be sampled directly, but we have a *proposal distribution* $Q(\mathbf{x})$ which is easy to be sampled from. Assume that we

also know a constant value c such that

$$cQ(\mathbf{x}) > P(\mathbf{x}) \tag{2.44}$$

Then the rejection sampling is performed according to the following steps:

(1) Sample x from $Q(\mathbf{x})$.

(2) Generate μ from $[0, cQ(x)]$.

(3) If $\mu > P(x)$, reject it, else keep it as one sample.

(4) Iterate the above steps until enough samples have been generated.

The main problem for reject sampling is that in high dimensional case, the c would be very large and the rejection rate would be very high, thus it may be too time-consuming to be a practical method.

Importance sampling. The idea of importance sampling is similar to the rejection sampling, but it is more efficient in that it will generate *weighted sample* set instead of the *unweighted sample* set from reject sampling.

The steps for importance sampling is as follows: instead of sampling from $P(\mathbf{x})$ directly, we sample x from the so-called *importance function* $Q(\mathbf{x})$ which is easy to sample from. Then we assign a compensation weight to x, i.e.,

$$\omega = \frac{P(x)}{Q(x)} \tag{2.45}$$

The problem for importance sampling, especially in the high dimensional case, is that the *importance function* $Q(\mathbf{x})$ must be as close to $P(x)$ as possible to efficiently achieve a satisfactory sample set.

2.4.3.2. Markov chain Monte Carlo. In the literature of applied statistics, there have been a family of well studied algorithms to sample from complex probabilistic distributions. It usually involves the construction of a Markov chain whose stationary distribution is the probabilistic distribution from which we want to sample. Then we can generate samples by simulating the transitions of the Markov chains. After discarding the first certain number of samples, which we call the samples in the *burn-in* stage, the sample set will represent the target distribution very well. This family of sampling algorithms are called *Markov chain Monte Carlo* (MCMC).

To guarantee that the stationary distribution of a Markov chain converges to the desired distribution, one sufficient condition for that is the so-called *detailed balance*, i.e.,

$$Q(x'|x)P(x) = Q(x|x')P(x'), \tag{2.46}$$

where $P(\mathbf{x})$ is the target distribution and $Q(\mathbf{x}'|\mathbf{x})$ is the transition probability of the Markov chain. Please refer to [**34**] for a broad discussion of the MCMC algorithms.

Metropolis-Hasting algorithm. The Metropolis-Hasting algorithm [**79**] might be one of the most generally applied MCMC algorithms. Suppose the current state of the Markov chain is x, we then generate a proposal transition sample x' from a proposal distribution $T(\mathbf{x}'|\mathbf{x})$, then with probability

$$Q(x'|x) = \min\left(1, \frac{T(x'|x)P(x)}{T(x|x')P(x')}\right) \tag{2.47}$$

the Markov chain transits to x', otherwise it stays in x. It can be easily shown that the transition probability defined in Eq. 2.47 guaranttees the detailed balance condition. Note that there is no precondition for the proposal distribution $T(\mathbf{x}'|\mathbf{x})$ used, but a good one will

enable the Markov chain to converge more quickly (higher mixing rate) and thus facilitate the sampling process. While a bad one will result in a slowly converged Markov chain and thus slow down the whole sampling process.

Gibbs sampling. Gibbs sampling [**33**] takes a different strategy to formulate the Markov chain for sampling. At every time instant, the transition probability of the Markov chain is the conditional probability of one random variable of the system given the state of all the other random variables in the probabilistic system. Let the set of all the random variables in the probabilistic system be $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_{\mathcal{L}}\}$ and the joint probability be $P(\mathbf{X})$, suppose that at one time instant t, the state of the Markov chain is at $X_t = \{x_1, \ldots, x_{\mathcal{L}}\}$, then the Gibbs sampling perform the following operations to generate the samples:

(1) Random choosing a $\mathbf{x}_i, 1 \leq i \leq \mathcal{L}$ with uniform probability for transition.

(2) Sample x_i' from the conditional probability

$$P(\mathbf{x}_i | x_1, \ldots, x_{i-1}, x_{i+1} \ldots x_{\mathcal{L}}) \tag{2.48}$$

(3) The state will be transited to $X_{t+1} = \{x_1, \ldots, x_{i-1}, x_i', x_{i+1} \ldots x_{\mathcal{L}}\}$.

One advantage of the Gibbs sampling algorithm over the Metropolis-Hasting algorithm is that the state of the Markov chain will always be transited at each time instant, while it may stay in one state for a very long time if a bad proposal distribution is chosen in the Metroplolis-Hasting algorithm. Thus, the Gibbs sampling algorithm generally achieves higher mixing rate and thus faster convergence for sampling.

Slice sampling. Slice sampling [**84**] is a newly developed MCMC algorithm. The basic idea is originated from the observation that to sample from a univariate probabilistic distribution, we can sample uniformly from the region under the curve of the probabilistic density

function, and then take into account just the horizontal coordinates of these samples. This idea could be extended directly to the multi-variate case by sampling uniformly under the plot of the multi-variate probabilistic distribution.

Let the set of all the random variables in the probabilistic system be $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_{\mathcal{L}}\}$, and the joint probability distribution be $P(\mathbf{X})$, which may not even have been normalized. Then what the slice sampling algorithm does is to introduce an auxiliary real valued random variable $\mathbf{y}$, whose joint probability with $\mathbf{X}$ is a uniform distribution over the volume under the manifold defined by $P(\mathbf{x})$, i.e.,

$$g(\mathbf{X}, \mathbf{y}) = \begin{cases} \frac{1}{C_{\mathcal{Q}}} & 0 < \mathbf{y} < P(\mathbf{X}) \\ 0 & Otherwise \end{cases}, \tag{2.49}$$

where $C_{\mathcal{Q}} = \int_{\mathbf{X}} P(\mathbf{X}) d\mathbf{X}$. Then the steps of the slice sampling algorithm to transit from the current state $X_t = \{x_1, \ldots, x_{\mathcal{L}}\}$ of the Markov chain is as follows [**84**]:

(1) Draw a real value y, uniformly from $(0, P(X_t))$ to define the slice region $\mathcal{S}$.

(2) Find a super-rectangle $\mathcal{T} = (\mathbf{L}_1, \mathbf{R}_1) \times (\mathbf{L}_2, \mathbf{R}_2) \times \ldots \times (\mathbf{L}_{\mathcal{L}}, \mathbf{R}_{\mathcal{L}})$, which contains at least most part of the slice region $\mathcal{S}$.

(3) Draw the new state X_{t+1}, uniformly from part of the slice inside this rectangle, i.e., from $\mathcal{T} \cap \mathcal{S}$.

A detailed discussion about the convergence of the sliced sampling algorithm can be found in [**84**].

2.4.3.3. Sequential Monte Carlo. Sequential Monte Carlo algorithm [**24**] refers to the Monte Carlo method utilized to perform the Bayesian inference in dynamic probabilistic systems. In the computer vision literature, the most successful application of the sequential

Monte Carlo algorithm would be the CONDENSATION algorithm [**52**]. Please refer to [**24**] and [**52**] for more details about how to formulate a sequential Monte Carlo algorithm.

2.4.4. Miscellaneousness

2.4.4.1. Laplace method. Just as we have mentioned in Sec. 2.2.1, it is the general difficulty in evaluating the partition function or the normalization constant that confronts probabilistic inference. Laplace's method is an approximate algorithm in evaluating it. The idea behind it is quite simple. Suppose an unnormalized probability density $P_{\mathbf{u}}(\mathbf{x}), \mathbf{x} \in \mathcal{R}^K$, we need to evaluate the normalization constant

$$\mathbf{Z}_{\mathbf{u}} = \int_{\mathbf{x}} P_{\mathbf{u}}(\mathbf{x}) d\mathbf{x}. \tag{2.50}$$

Assume $P_{\mathbf{u}}(\mathbf{x})$ has a peak at x_0, then we expand the $\ln P_{\mathbf{u}}(\mathbf{x})$ using Taylor's expansion,

$$\ln P_{\mathbf{u}}(\mathbf{x}) \approx \ln P_{\mathbf{u}}(x_0) - \frac{(\mathbf{x} - x_0)^T A(x_0)(\mathbf{x} - x_0)}{2}, \tag{2.51}$$

where $A(x_0)$ is the Hessian matrix of $\ln P_{\mathbf{u}}(\mathbf{x})$ at $\mathbf{x} = x_0$. Then we have

$$P_{\mathbf{u}}(\mathbf{x}) \approx P_{\mathbf{u}}(x_0) \exp\left(-\frac{(\mathbf{x} - x_0)^T A(x_0)(\mathbf{x} - x_0)}{2}\right), \tag{2.52}$$

and thus

$$\mathbf{Z}_{\mathbf{u}} \approx P_{\mathbf{u}}(x_0) \sqrt{\frac{(2\pi)^K}{\det A(x_0)}}. \tag{2.53}$$

2.4.5. Model selection and model scoring criteria

2.4.5.1. Bayesian information criterion (BIC) and minimum description length (MDL). The Bayesian information criterion (BIC) or Schwarz criterion [**92, 93**] provides

us with a principled criterion to score a model. Although it is called Bayesian information criterion, it is actually neither Bayesian nor information theoretic [**38**,**121**]. It can be derived from the Laplace method. For a probabilistic system **ps**, given the observed random state $Z = \{z_1, \ldots, z_n\}$, the goodness of a model $\mathcal{H}$ with the d *effective parameters* denoted as $\mathbf{\Theta}$ could be evaluated by the BIC, i.e.,

$$BIC = -2 \ln P(Z|\hat{\theta}_{ML}, \mathcal{H}) + d \ln n, \tag{2.54}$$

where $\hat{\theta}_{ML}$ is the ML estimation of $\mathbf{\Theta}$ and n is the sample size. The smaller the BIC, the better the model. As we can easily observe, the subtracted quantity on the right side of Eq. 2.54 is exactly the minimum description length (MDL) penalty. Therefore it will be in favor of models with less number of effective parameters, given that the two models can explain the observation equally.

2.4.5.2. Akaike information criterion (AIC). Another widely adopted information criterion for model selection is the Akaike information criterion (AIC) [**1**]. It deserves the name of information criterion because it can be strictly derived based on information theory. Still for a probabilistic system **ps**, given the observed random state Z, the goodness of a model $\mathcal{H}$ with the *effective parameters* $\mathbf{\Theta}$ can be evaluated by the AIC, i.e.,

$$AIC = -2 \ln P(Z|\hat{\theta}, \mathcal{H}) + 2d, \tag{2.55}$$

where $\hat{\theta}$ is the ML estimation of $\mathbf{\Theta}$. The smaller the AIC, the better the model. We could easily observe that the AIC is also penalized by MDL.

2.5. Conclusion remarks

In this chapter, we present a brief yet comprehensive reviews of the techniques of probabilistic inference on graphical models, which is a powerful and principled means of uncertainty reasoning and empirical learning. Firstly, we discuss the three basic problems in probabilistic inference, namely *latent variable inference*, *parameter estimation* and *model selection*. Then, we introduce different types of graphical models including *Bayesian networks*, *Markov networks*, and *factor graphs* along with the discussions of their mathematic equivalences. The correspondences of the three basic problems under graphical models then become *latent variable inference*, *parameter learning* and *structure learning*. After that, we extensively discuss both exact and approximate probabilistic inference algorithms, as well as the criteria for model selections (*structure learning*). With the background technologies introduced in this chapter, we will proceed, in the following chapters, to develop novel algorithms based on graphical models and probabilistic variational analysis to address the fundamental challenges in visual analysis of complex motion.

CHAPTER 3

Mean field variational analysis for articulated body tracking

Articulated body motion composes one of the most important types of complex motions. Using it as a concrete example, this chapter presents a collaborative approach [**126**] to scalable analysis of vision based complex motion analysis. As a matter of fact, the proposed collaborative approach has also been applied to analyze other types of complex motions, including the complex deformation [**45**, **46**, **133**], as well as motions of multiple identical targets [**137**].

3.1. Introduction

Tracking articulated motion in video is an important problem, especially when the research of video-based human sensing has been advocated to achieve such emerging applications such as non-invasive perceptual human computer interfaces [**11**, **127**], intelligent video surveillance [**36**,**124**], gait analysis [**114**,**119**], automatic hand gesture recognition [**128**,**129**] and automatic video footage annotation [**14**], etc.

The problem involves the localization and identification of a set of linked but articulated limbs. Inheriting all the difficulties from single object tracking, the problem of tracking articulated body has to tackle some special challenges. One of these is the complexity incurred by the degrees of freedom of the articulated body.

Different from multiple target tracking where the motion of each target is usually independent of the others, the physical links among different limbs impose motion constrains

upon them. In other words, the motion of each limb must be spatially coherent with the others, which is reinforced by the kinematic structure of the articulated limbs. We can have an intuitive comparison of these two cases by the configuration space which is the joint motion space of the set of limbs. If the motions of limbs are independent, the configuration space will enjoy a nice property that the motion of each limb stays in a manifold that is orthogonal to the manifolds corresponding to the other limbs. Thus, independent trackers can be used to track independent multiple targets and the complexity is almost linear w.r.t. the number of targets. However, when the limbs are physically linked, the configuration space will not have such a nice orthogonality and factorization property. Thus, the high dimensionality seems unavoidable, which is generally associated with the exponential increase of computation due to the curse of dimensionality.

Various approaches have been investigated to alleviate the computation complexity caused by high dimensionality, such as dynamic programming [**27**, **109**], annealed sampling [**22**], partitioned sampling [**76**, **77**], eigen-space tracking [**8**], hybrid Monte Carlo filtering [**15**], covariance scaled sampling [**108**], etc., to name a few.

Different from these approaches, in this chapter, we propose a novel solution based on a dynamic Markov network [**45**, **46**, **126**] and a mean field variational analysis. The proposed dynamic Markov network encodes the spatial coherence of different limbs in an undirected graphical model associated with the image observation processes, thus the model serves as a generative model for the articulated motion. We perform the Bayesian inference based on a variational mean field method, by which tight approximation may be achieved while the computational complexity is significantly reduced. At each time instance, the mean field solution is achieved through Monte Carlo simulation. Based on that, we design a mean

field sequential Monte Carlo for articulated body tracking. Extensive experiments show the effectiveness and efficiency of the proposed approach.

The remainder of this chapter is organized as follows: related work is summarized in Sec. 3.2. Then, a distributed probabilistic representation of articulated body is presented in Sec. 3.3 based on Markov networks. After that, we present the general idea of mean field variational analysis for Bayesian inference in Sec. 3.4 followed by the Monte Carlo implementation of the mean field fixed-point iteration called mean field Monte Carlo (MFMC) [**45**, **46**, **126**] in Sec. 3.5. In the context of dynamics, we adopt a dynamic Markov network to model the articulated motion at each time instant. The Bayesian inference is performed by a sequential version of the MFMC algorithm, which is called sequential MFMC. They are all presented in Sec. 3.6. Extensive experimental results are reported in Sec. 3.7. Finally we conclude this chapter in Sec. 3.8.

3.2. Related work

There is a substantial literature on articulated motion analysis, and many different approaches have been investigated. For all these methods, three important issues should be addressed: the representations for articulated objects, the computational paradigms, and the way of reducing computation.

There can be two typical representations for articulated object. One employs the joint angles [**14**, **77**, **97**, **131**], which is in nature a centralized representation. While the other uses the collection of the motion of all the limbs, e.g., the cardboard person [**60**], the decentralized probabilistic model based on Markov network [**43**, **126**], the loose-limbed model [**105**, **106**], and tree structured model [**27**, **94**], to list a few.

Of course, the centralized joint angle representation is non-redundant and reflects the degrees of freedom of the articulated motion directly, while the second one is highly redundant. The centralized representation usually results in a very high dimensional parametrization. Since there are complex motion constraints, it may be possible to learn a lower dimensional manifold to characterize the articulated motion [**104**, **131**]. However, the intrinsic dimensionality of the learned manifold may still be quite high. In this case, the motion analysis problem can be posed as an unconstrained optimization problem in a high dimensional space. On the other hand, if the articulated motion is redundantly described by the individual motion of the subparts, each subpart may be solved individually, and then projected to the constrained space which reinforces the spatial coherence among them. Thus, it corresponds to a constrained optimization problem. By taking advantage of the structure of the configuration space resulted from such a redundant representation, efficient solutions can be found as in this chapter.

There are mainly two different computational paradigms for articulated motion analysis: the deterministic approach usually formulates the problem as a parameter estimation problem [**10**,**60**,**97**], and the solution is usually provided by some nonlinear optimization methods. While the probabilistic approach formulates it as a Bayesian inference problem [**22**,**105**,**126**], and the solution is provided by recovering the motion posteriors sequentially at each time instant. Due to the non-Gaussian densities which commonly exist in a probabilistic formulation [**9**, **52**], closed-form implementation of the Bayesian inference is mostly intractable and thus it is usually performed by Monte Carlo simulation. However, both approaches are confronted by the high dimensionality. More specifically, for the deterministic approach, the

optimization needs to be performed in a very high dimensional parametric space which is confronted by the numerous local optima. As for the probabilistic approach, the computational cost of a Monte Carlo algorithm may increase exponentially with the dimensionality [**75**].

Therefore, it is crucial to reduce the computation. Numerous techniques have been proposed to improve the efficiency of the probabilistic approach. For example, a multiple hypothesis tracking algorithm is proposed [**14**], which only keeps the salient modes of the motion posteriors for more efficient Monte Carlo simulation. Partitioned sampling is in the spirit of coordinate descent and preforms the sampling in a hierarchical fashion [**76**, **77**]. Low dimensional manifold could be learned from the natural hand motion to reduce the dimensionality [**131**]. In [**105**, **106**], the non-parametric belief propagation algorithm [**51**, **110**] is applied on the loose-limbed model to achieve the Bayesian inference of the articulated body motion. Different from these methods, this chapter presents a mean field Monte Carlo (MFMC) algorithm in which a set of low dimensional particle filters interact with one another to solve a high dimensional problem collaboratively.

3.3. The representation of an articulated body

We denote the motion of each individual limb by $\mathbf{x}_i$, which can be the parameters of an affine motion. The motion of an articulated body is the concatenation $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_M\}$. Certainly, it is highly redundant. The image observation associated with $\mathbf{x}_k$ is denoted by $\mathbf{z}_k$, which can be the detected edges of the shape contours of the limbs. The collective image observations of the entire articulated body is $\mathbf{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_M\}$. An important task is to infer the posterior $P(\mathbf{X}|\mathbf{Z})$.

As shown in Fig. 3.1, a mixture of undirected and directed graphical model can be used to characterize the generative process. The hidden layer is an undirected graph $G_x =$

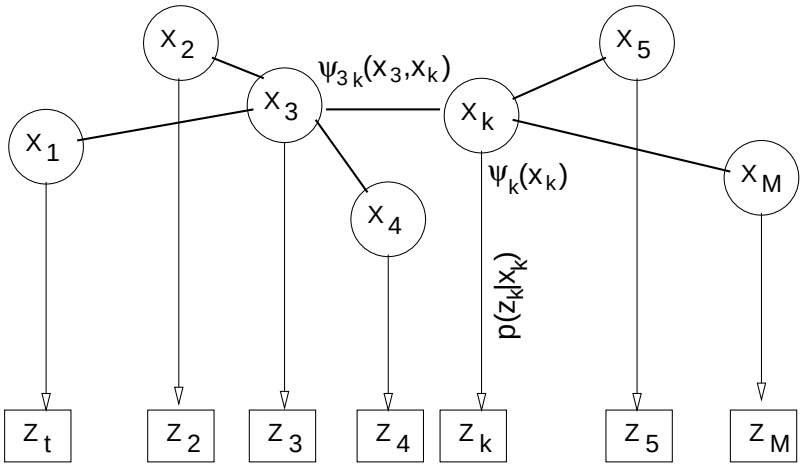


Figure 3.1. The Markov Network for an articulated body.

$\{V, E\}$, representing the spatial coherence constraints among different articulated limbs. Obviously, different limbs are not independent, and each individual limb only interacts with its neighborhood parts. We denote the neighborhood limbs of i by $\mathcal{N}(i)$. Clearly, it is a Markov network. In addition, each individual limb is associated with its observation and the conditional likelihood distribution $P(\mathbf{z}_i|\mathbf{x}_i)$ is represented by a directed link.

Given the undirected graph of $\mathbf{X}$, $P(\mathbf{X})$ can be modeled as a Gibbs distribution and can be factorized as:

$$P(\mathbf{X}) = \frac{1}{C_c} \prod_{c \in \mathcal{C}} \psi_c(X_c) \tag{3.1}$$

where c is a clique in the set of cliques $\mathcal{C}$ of the undirected graph, X_c is the set of hidden nodes associated with the clique c, $\psi_c(X_c)$ is the potential function of this clique, and C_c is a normalization term or the partition function. Although C_c is difficult to compute, we do not compute it directly, since a Monte Carlo method will be used as shown in later sections. The model accommodates two types of cliques: the first order clique, i.e., $i \in \mathcal{C}^1 = V$, and second order clique, i.e., $(i, j) \in \mathcal{C}^2 = E$, where $\mathcal{C} = \mathcal{C}^1 \bigcup \mathcal{C}^2$. The associated ψ is denoted

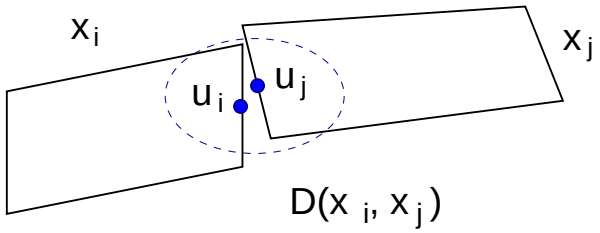


Figure 3.2. The constraint of two articulated parts.

by ψ_i and ψ_{ij}, respectively. Thus, Eq. 3.1 can also be written as:

$$P(\mathbf{X}) = \frac{1}{C_c} \prod_{(i,j)\in\mathcal{C}^2} \psi_{ij}(\mathbf{x_i}, \mathbf{x_j}) \prod_{i\in\mathcal{C}^1} \psi_i(\mathbf{x_i}) \tag{3.2}$$

where $\psi_i(\mathbf{x_i})$ provides a local prior for $\mathbf{x}_i$, and $\psi_{ij}(\mathbf{x_i}, \mathbf{x_j})$ presents the constraints between the neighborhood nodes $\mathbf{x}_i$ and $\mathbf{x}_j$. In other words, $\psi_i(\mathbf{x_i})$ represents a prior for the i-th part, while $\psi_{ij}(\mathbf{x_i}, \mathbf{x_j})$ reinforces the spatial coherence constraints between the i-th part and the j-th part. As a specific example, it can be modeled as:

$$\psi_{ij}(\mathbf{x_i}, \mathbf{x_j}) \propto e^{-\frac{1}{2}D(\mathbf{x}_i,\mathbf{x}_j)^T\Sigma^{-1}D(\mathbf{x}_i,\mathbf{x}_j)} \tag{3.3}$$

where $D(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{u}_i(\mathbf{x}_i) - \mathbf{u}_j(\mathbf{x}_j)$, and $\mathbf{u}_i(\mathbf{x}_i)$ and $\mathbf{u}_j(\mathbf{x}_j)$ are shown in Fig. 3.2. Here we must emphasize that the zero mean Gaussian prior is a very weak prior, which only captures the connectivity of the neighborhood limbs. The reason we adopt it is that our goal is to analyze arbitrary articulated body motion instead of specific ones. More complex spatial coherence potential functions may be learned for more specific stylized articulated motions. Given a $\mathbf{x}_i$, its local observation $\mathbf{z}_i$ is independent of other articulated parts. Thus, we have:

$$P(\mathbf{Z}|\mathbf{X}) = \prod_{i=1}^{n} p_i(\mathbf{z_i}|\mathbf{x_i}). \tag{3.4}$$

The problem of great interest is to infer the posterior $P(\mathbf{x}_i|\mathbf{Z})$. An intuition is that the posterior of $\mathbf{x}_i$ should be affected by three factors: its local prior ψ_i, its local evidence $\mathbf{z}_i$, and the spatial coherence constraints reinforced by its neighborhood through ψ_{ij}. This intuition will become clearer in Sec. 3.4. Since the exact analysis of such a model is complicated and involves heavy computation, it is more plausible to have an approximate but efficient solution.

3.4. Mean field variational analysis

Variational analysis provides a principled method for approximate Bayesian inference [**6**, **55**, **59**, **123**]. The core idea of variational approximation is to find a variational distribution $Q(\mathbf{X})$ to approximate the posterior distribution $P(\mathbf{X}|\mathbf{Z})$, such that the following cost function is minimized, i.e.,

$$J(Q) = \log P(\mathbf{Z}) - KL\left(Q(\mathbf{X}) \| P(\mathbf{X}|\mathbf{Z})\right). \tag{3.5}$$

It is easy to figure out that maximizing $J(Q)$ is equivalent to minimizing $KL(Q(\mathbf{X})||P(\mathbf{X}|\mathbf{Z}))$ since $P(\mathbf{Z})$ is in fact a constant. Selecting a good class of variational distributions Q would largely ease the difficulties of optimization, but it requires substantial creativity [**59**]. Here, we adopt a fully factorized form:

$$Q(\mathbf{X}) = \prod_{i}^{M} Q_i(\mathbf{x}_i) \tag{3.6}$$

where $Q_i(\mathbf{x_i})$ only relies on $\mathbf{x}_i$. Then we have

$$J(Q) = \log P(\mathbf{Z}) - \oint_{\mathbf{X}} \prod_j Q_j(\mathbf{x}_j) \log \left(\frac{\prod_j Q_j(\mathbf{x}_j)}{P(\mathbf{X}|\mathbf{Z})} \right) d\mathbf{X} \tag{3.7}$$

$$= \sum_j H_j(Q_j(\mathbf{x}_j)) + \int_{\mathbf{x}_i} Q_i(\mathbf{x}_i) E_Q \left[\log P(\mathbf{X}, \mathbf{Z})|\mathbf{x}_i\right] d\mathbf{x}_i, \tag{3.8}$$

where

$$H_j(Q_j(\mathbf{x}_j)) = -\int_{\mathbf{x}_j} Q_j(\mathbf{x}_j)\log Q_j(\mathbf{x}_j)d\mathbf{x}_j \tag{3.9}$$

is the entropy of the distribution $Q_j(\mathbf{x}_j)$ and

$$E_Q\left[\log P(\mathbf{X},\mathbf{Z})|\mathbf{x}_i\right] = \oint_{\{\mathbf{x}_j\}\backslash\mathbf{x}_i}\prod_{\{\mathbf{j}\}\backslash\mathbf{i}} Q_j(\mathbf{x}_j)\log P(\mathbf{X},\mathbf{Z})d\mathbf{X}. \tag{3.10}$$

The problem of Bayesian inference becomes a constrained optimization problem, i.e.,

$$\boxed{\begin{array}{ll}\text{Maximize} & J(Q)\\ s.t. & \int_{\mathbf{x}_i} Q_i(\mathbf{x}_i)d\mathbf{x}_i = 1, \quad for \quad i = 1\ldots M\end{array}} \tag{3.11}$$

It is solved by formulating the following Lagrangian multiplier, i.e.,

$$J^*(Q) = J(Q) + \sum_i \lambda_i\left(\int_{\mathbf{x}_i} Q_i(\mathbf{x}_i)d\mathbf{x}_i - 1\right). \tag{3.12}$$

To optimize $J^*(Q)$, take the variation of $J^*(Q)$ w.r.t. each $Q_i(\mathbf{x}_i)$ and the derivative of $J^*(Q)$ w.r.t. each λ_i and setting them to zero, we obtain the following set of Euler equations, i.e.,

$$\begin{cases} -\log Q_i(\mathbf{x}_i) - 1 + E_Q[\log P(\mathbf{X},\mathbf{Z})|\mathbf{x}_i] + \lambda_i = 0 \\ \int_{\mathbf{x}_i} Q_i(\mathbf{x}_i)d\mathbf{x}_i - 1 = 0 \end{cases}. \tag{3.13}$$

Solve this equation set, we easily obtain

$$\begin{cases} Q_i(\mathbf{x}_i) = \exp\left(\lambda_i - 1\right)\exp\left(E_Q[\log P(\mathbf{X},\mathbf{Z})|\mathbf{x}_i]\right) \\ \lambda_i = 1 - \log\left(\int_{\mathbf{x}_i}\exp\left\{E_Q[\log P(\mathbf{X},\mathbf{Z})|\mathbf{x}_i]\right\}\right) \end{cases}. \tag{3.14}$$

We thus obtain a set of fixed point equations, i.e., for each $1 \le i \le M$,

$$Q_i(\mathbf{x}_i) = \frac{1}{C_i}\exp\left\{E_Q[\log P(\mathbf{X},\mathbf{Z})|\mathbf{x}_i]\right\} \tag{3.15}$$

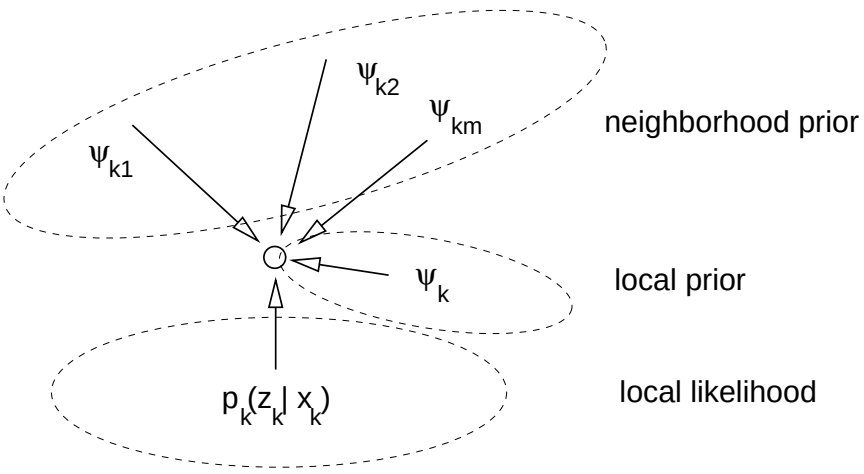


Figure 3.3. Three factors affecting the updating of $Q(\mathbf{x}_k)$.

where C_i is the partition function for normalization. The iterative updating of $Q_i(\mathbf{x}_i)$ will monotonically decrease the KL divergence, and eventually reach an equilibrium. These fixed-point equations are called *mean field equations*.

Moreover, the factorization of $P(\mathbf{X})$ in Eq. 3.2 and $P(\mathbf{Z}|\mathbf{X})$ in Eq. 3.4 enable further simplification of the mean field equations in Eq. 3.15. It is easy to show that:

$$Q_i(\mathbf{x}_i) = \frac{1}{C_i'} p_i(\mathbf{z}_i|\mathbf{x}_i)\psi_i(\mathbf{x}_i)M_i(\mathbf{x}_i), \tag{3.16}$$

where

$$M_i(\mathbf{x}_i) = \exp\left\{\sum_{k\in\mathcal{N}(i)} \int_{x_k} Q_k(\mathbf{x}_k)\log\psi_{ik}(\mathbf{x}_i,\mathbf{x}_k)\right\}, \tag{3.17}$$

where C_i' is a constant, and $\mathcal{N}(i)$ is the neighborhood of limb i. From Eq. 3.16, the intuition stated at the end of Sec. 3.3 is more pronounced, i.e., the variational belief of a limb $\mathbf{x}_i$ is determined by three factors: the local conditional likelihood $p_i(\mathbf{z}_i|\mathbf{x}_i)$, the local prior $\psi_i(\mathbf{x}_i)$, and the beliefs from the neighborhood limb $\mathbf{x}_{\mathcal{N}(i)}$ (we call it neighborhood prior). This is illustrated in Fig. 3.3.

Thus,we can treat the term $p_i(\mathbf{z}_i|\mathbf{x}_i)\psi_i(\mathbf{x}_i)$ as an analogue to the local belief, and treat the term $M_i(\mathbf{x}_i)$ as an analogue to the "message" propagated through the nearby subpart of $\mathbf{x}_i$ in the belief propagation algorithm [**28**], but the computation of $M_i(\mathbf{x}_i)$ here is easier. In addition, we can clearly see from these equations that the computation is significantly reduced by avoiding multi-dimensional integrals, since Eq. 3.16 involves only single integral.

3.5. Mean field Monte Carlo (MFMC)

In this section, we propose a Monte Carlo method to implement the mean field iterations as discussed in Sec. 3.4. We call this method *mean field Monte Carlo* (MFMC).

Once the mean field iterations converge, then the set of optimal variational distributions $Q_i(\mathbf{x}_i)$, where $i = 1, \ldots, M$, is obtained and can be treated as the optimal approximation to the posterior density $P(\mathbf{x}_i|\mathbf{Z})$. To make the presentation clear, here we use a 2-link body as an example. Without loss of any generality, we use i and j to index the two linked limbs, and we use k to index the mean field iteration. At the $k-1$-th iteration, for each limb, a set of particle is maintained to represent the variational distribution, i.e.,

$$\begin{aligned} Q_i^{k-1}(\mathbf{x}_i) &\sim \{s_i^{(n)}(k-1), \pi_i^{(n)}(k-1)\}_{n=1}^N \\ Q_j^{k-1}(\mathbf{x}_j) &\sim \{s_j^{(n)}(k-1), \pi_j^{(n)}(k-1)\}_{n=1}^N \end{aligned} \tag{3.18}$$

where s and π denote the sample and the weight respectively. Then at the next iteration, we perform the following steps according to Eq. 3.16:

(1) Sampling local prior $\psi_i(\mathbf{x}_i)$ for $\{s_i^{(n)}(k), 1\}_{n=1}^N$;

(2) Calculating the "message" from j:

$$m_i^{(n)} = \sum_{t=1}^{N} \pi_j^{(n)}(k-1) \log \psi_{ij}(s_i^{(n)}(k), s_j^{(t)}(k-1)). \tag{3.19}$$

(3) Performing observation for each particle $s_i^{(n)}(k)$,

$$w_i^{(n)} = P(z_i | s_i^{(n)}(k)). \tag{3.20}$$

(4) Re-weighting the particles by:

$$\pi_i^{(n)}(k) = e^{m_i^{(n)}} \times w_i^{(n)}. \tag{3.21}$$

and normalize to produce $\{s_i^{(n)}(k), \pi_i^{(n)}(k)\}$.

(5) Performing the same steps for j according to Eq. 3.16. And then increase k for next mean field updating.

After the k-th iteration, we end up with:

$$Q_i^k(\mathbf{x}_i) \sim \{s_i^{(n)}(k), \pi_i^{(n)}(k)\}_{n=1}^N$$

$$Q_j^k(\mathbf{x}_j) \sim \{s_j^{(n)}(k), \pi_j^{(n)}(k)\}_{n=1}^N$$

After several iterations, the distribution will reach an equilibrium. For a limb which is linked to multiple limbs, the only difference is in the 2nd step of calculating "messages",

$$m_i^{(n)} = \sum_{j \in \mathcal{N}(i)} \sum_{t=1}^{N} \pi_j^{(n)}(k-1) \log \psi_{ij}(s_i^{(n)}(k), s_j^{(t)}(k-1)). \tag{3.22}$$

which sums over all "messages" passed from the neighbors $\mathcal{N}(i)$ (i.e., the Markov blanket) of $\mathbf{x}_i$.

Since s_i describes the hypothesis motion of limb i, its image observation z_i should be a function of s_i, i.e., $P(\mathbf{z}_i|\mathbf{x}_i)$ is in fact evaluated by $P(z_i(s_i)|s_i)$. Since $P(z_i(s_i)|s_i)$ will be used to re-weight the belief (or the posterior density) of $\mathbf{x}_i$, the locations of the particles $\{s_i^{(n)}\}$ will affect the faith of approximating the belief by the set of particles if the ratio of valid particles is not satisfactory (meaning that a small portion of the particles dominates the re-weighting). To enhance the ratio of valid particles, we use importance sampling technique [**74**] to place the particles to "better" locations.

The only modification on the above mean field Monte Carlo (MFMC) is on the first step: instead of sampling the local prior $\psi_i(\mathbf{x}_i)$ directly to produce $\{s_i^{(n)}, \frac{1}{N}\}_{n=1}^N$, we generate samples $\{s_i^{(n)}, \frac{1}{N}\}_{n=1}^N$ from an importance density $g(\mathbf{x}_i)$. After weight compensation, the set of re-weighted particle is still a properly weighted set for the density $\psi_i(\mathbf{x}_i)$, i.e.,

$$\psi_i(\mathbf{x}_i) \sim \{s_i^{(n)}, \frac{\psi_i(s_i^{(n)})}{g(s_i^{(n)})}\}_{n=1}^N. \tag{3.23}$$

The selection of importance density may be arbitrary, as long as it can provides beneficial information. Here we give an specific example by using a two-link (where i and j are connected limbs). To generate samples for $\psi_i(\mathbf{x}_i)$, we find the means $\bar{s}_i$ and $\bar{s}_j$ from the two particle sets. After identifying the point $\bar{u}_j$ on $\bar{s}_j$ and the median axis $\bar{L}_i$ of $\bar{s}_i$ (see Fig. 3.4), we sample $u_i^{(n)}$ from $\mathcal{G}(u_i : \bar{u}_j, \Sigma_u)$, and $L_i^{(n)}$ from $\mathcal{G}(L_i : \bar{L}_i, \Sigma_L)$, where $\mathcal{G}$ is a Gaussian.

Then the sample $s_i^{(n)}$ is produced by $(L_i^{(n)}, u_j^{(n)})$, and the importance density is:

$$g(\mathbf{x}_i) = \mathcal{G}(u_i : \bar{u}_j, \Sigma_u)\mathcal{G}(L_i : \bar{L}_i, \Sigma_L). \tag{3.24}$$

For limbs which are linked to multiple limbs, we can build one such a Gaussian from each of its neighbors. Then a Gaussian mixture with equal weights for each of the Gaussian

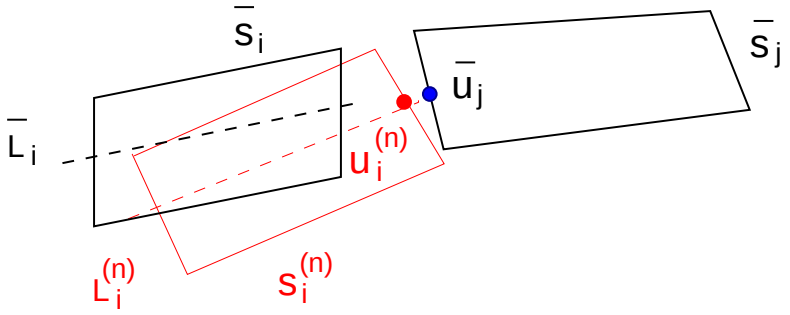


Figure 3.4. Importance density.

components can be constructed to form the importance function, i.e.,

$$g(\mathbf{x}_i) = \frac{1}{K} \sum_{j \in \mathcal{N}(i)} g_j(\mathbf{x}_i) \tag{3.25}$$

where K is the total number of neighbors of $\mathbf{x}_i$. The use of importance sampling techniques greatly enhances the robustness of the mean field Monte Carlo algorithms.

3.6. Dynamic Markov network and sequential mean field Monte Carlo

Sec. 3.4 and Sec. 3.5 describe the mean field approximation and mean field Monte Carlo at one time instance. They can be easily modified for tracking. When considering multiple time instances, the model becomes a dynamic Markov network, as shown in Fig. 3.5. Denote the collection of observations by $\underline{\mathbf{Z}}_t = \{\mathbf{Z}_1, \ldots, \mathbf{Z}_t\}$. Tracking algorithms aim at inferring $P(\mathbf{X}_t|\underline{\mathbf{Z}}_t)$ by knowing $P(\mathbf{X}_{t-1}|\underline{\mathbf{Z}}_{t-1})$. It involves a density propagation process [**52**]. Once $\mathbf{X}$ consists of a number of articulated parts, the increase of dimensionality will incur exponential increase of computation. The advantage of mean field approximation is that it decouples different parts, and transforms the problem of exponential complexity to a simpler problem with close to linear complexity. The constraint reinforcement needs some computation as a cost, but it is not significant.

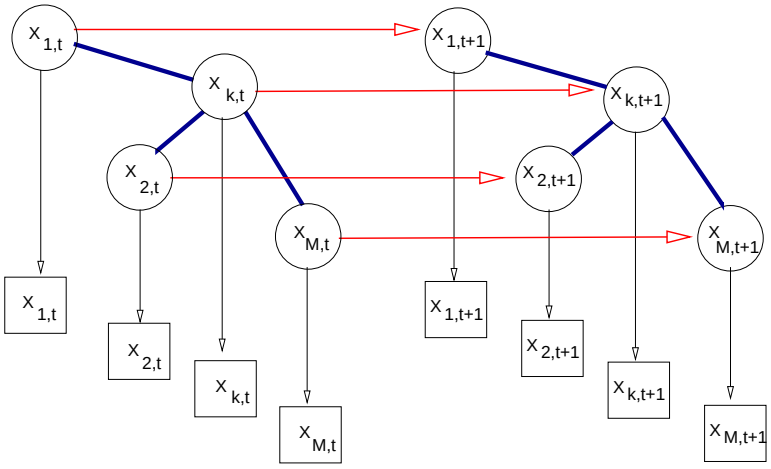


Figure 3.5. Dynamic Markov network for articulated body motion.

At time instance t, mean field approximation finds a variational distribution $Q_{i,t}(\mathbf{x}_{i,t})$ to approximate $P(\mathbf{x}_{i,t}|\underline{\mathbf{Z}}_t)$ for the i-th limb. The mean field equation can be written as:

$$Q_{i,t}(\mathbf{x}_{i,t}) = \frac{1}{C_i'} p_i(\mathbf{z}_{i,t}|\mathbf{x}_{i,t}) \times \int P(\mathbf{x}_{i,t}|\mathbf{x}_{i,t-1}) Q_{i,t-1}(\mathbf{x}_{i,t-1}) d\mathbf{x}_{i,t-1}$$
$$\times \exp\left\{ \sum_{k \in \mathcal{N}(i)} \int_{x_{k,t}} Q_{k,t}(\mathbf{x}_k) \log \psi_{ik}(\mathbf{x}_{i,t}, \mathbf{x}_{k,t}) \right\} \tag{3.26}$$

Comparing Eq. 3.26 to Eq. 3.16, we clearly observe that the prediction probabilistic density $\int P(\mathbf{x}_{i,t}|\mathbf{x}_{i,t-1}) Q_{i,t-1}(\mathbf{x}_{i,t-1}) d\mathbf{x}_{i,t-1}$ in Eq. 3.26 plays the same role as $\psi_i(\mathbf{x}_i)$ in Eq. 3.16. Thus, at time instance t, the variational belief of the i-th limb is also determined by three factors: the local evidence, the prediction prior from previous time frame, and the belief of the neighborhood limbs.

Therefore, the sequential mean field Monte Carlo can be obtained by modifying the mean field Monte Carlo algorithm in Sec. 3.5. At the first step, instead of sampling from $\psi_i(\mathbf{x}_i)$, we should sample the prediction prior instead. Suppose at $t-1$, $Q_{i,t-1}(\mathbf{x}_{i,t-1})$ is represented

by:

$$Q_{i,t-1}(\mathbf{x}_{i,t-1}) \sim \{s_{i,t-1}^{(n)}, \pi_{i,t-1}^{(n)}\}_{n=1}^{N}. \tag{3.27}$$

Then, we can use the following steps to replace the 1st step in the mean field Monte Carlo algorithm in Sec. 3.5:

1.a Re-sampling from $Q_{i,t-1}(\mathbf{x}_{i,t-1})$ for $\{\widetilde{s}_{i,t-1}^{(n)}, \frac{1}{N}\}_{n=1}^{N}$.

1.b $\forall\ \widetilde{s}_{i,t-1}^{(n)}$, sampling $s_{i,t}^{(n)}$ from $P(\mathbf{x}_{i,t}|\ \widetilde{s}_{i,t-1}^{(n)})$.

Impressive results are obtained and are reported in Sec. 3.7.

We have a rough comparison on the computational complexity of the proposed approach with the original CONDENSATION algorithm [**52**] with joint angle representation. Assume the articulated body consists of M limbs, each of which contribute one DoF, and assume a number of T particles are needed for tracking one limb. In addition, we assume when one more DoF is added, CONDENSATION needs $P \times T$ particles to work. Through our experiments, 10 is reasonable for P. In our mean field Monte Carlo, we denote the number of mean field iterations by K, which is 5 in our experiments. In both methods, the most intensive computation is on calculating image observation, while the extra computation induced by $M_i(\mathbf{x}_i)$ in Eq. 3.16 is negligible. Thus, the complexity of our method is $O(TKM)$, while CONDENSATION has $O(TP^{M-1})$ which is much higher than the proposed mean field Monte Carlo algorithm.

In addition, the proposed mean field Monte Carlo (MFMC) algorithm is also different from the partitioned sampling method, although both methods reduce the exponential complexity to close linear complexity. Partitioned sampling needs the independence assumption to decompose the dynamics, thus it works with the joint angle representation, while the proposed MFMC formulates the problem in a differen way. In addition, partitioned sampling

is a hierarchical search which is uni-directional (it may be revised to run back and forth, though), while MFMC is collaborative and iterative, since the fixed point is achieved by the bi-directional interactions among a set of low-dimensional particle sets.

3.7. Experiments on tracking articulated body

We perform extensive experiments on articulated body with different DoFs, and obtain impressive results as reported in this section.

3.7.1. Experimental setup

Our experiments mainly concern about 2D tracking. Thus we adopt a cardboard model where each limb in the articulated body is represented by a planar object, and thus the state of $\mathbf{x}_i$ is the parameters of a 2D affine transform. The motion model $P(\mathbf{x}_{i,t}|\mathbf{x}_{i,t-1})$ is a standard 2nd order constant acceleration model for each limb, which can be estimated online.

The observation model $P(\mathbf{z}_i|\mathbf{x}_i)$ is also an important factor in tracking. We use two types of visual cues: edge and intensity. We adopt the same method in CONDENSATION [**9**, **52**] for edge observation, where a set of independent measurement lines were used to measure the likelihood of detected edge points. In addition, since the articulated targets are human body parts and the skin or clothes on the body parts are similar, we also use the intensity clue and assume the distribution of the intensity of a limb be a Gaussian distribution. The mean and variance of the Gaussian density is trained for each individual limb with the manual initialization in the first frame.

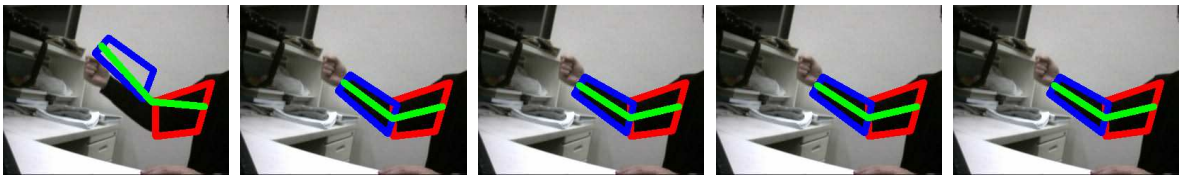


Figure 3.6. The first five iterations of MFMC on the (2-part) Arm sequence.

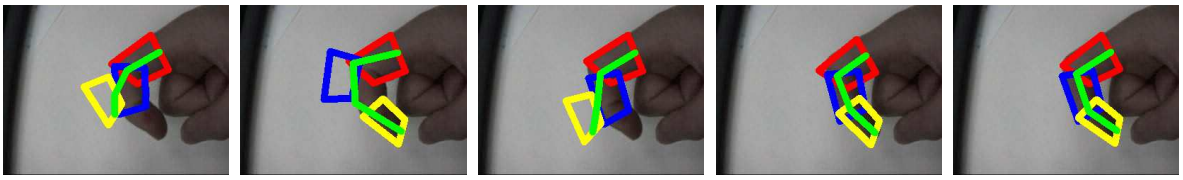


Figure 3.7. The first five iterations of MFMC on the (3-part) Finger sequence.

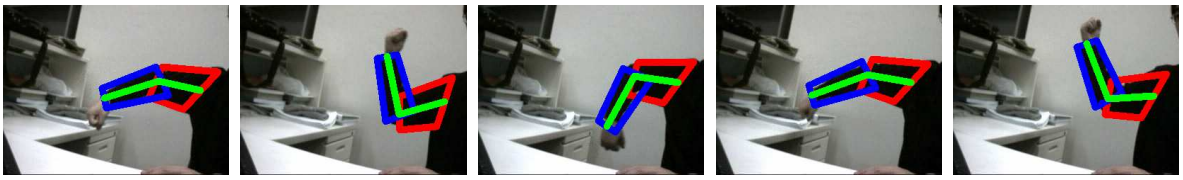


Figure 3.8. Mean field Monte Carlo (MFMC): tracking 2-part arm.

3.7.2. Results of MFMC iteration

To demonstrate that the mean field iterations do converge and function as expected, we collect the intermediate results on MFMC iterations. Two examples are shown in Fig. 3.6 and Fig. 3.7, which show the iteration process at a specific time instant on a 2-part arm sequence, and on a 3-part finger sequence, respectively. In both cases, the estimates of the first five iterations are shown. Before the iteration, the initial status is quite unpleasant. But after a couple of mean field iterations, the estimates settle down on the right positions as expected. From our experiments, most iterations converge in less than five times.

Figure 3.9. Multiple independent tracker (MiT): tracking 2-part arm.

3.7.3. Various articulated objects

To demonstrate the effectiveness, efficiency and scalability, we perform experiments on various articulated objects of difference DoFs, including a 2-part arm, 3-part finger, 6-part upper body, and 10-part full body.

The first test sequence is a 2-part arm, which consists of two limbs: upper arm and lower arm. The sequence consists of 441 frames. The lower arm presents larger motion than upper arm in the testing sequence. The MFMC algorithm performs excellently due to the constraint reinforcement. Sample frames are shown in Fig. 3.8[1].

We compare the results from MFMC with multiple independent trackers (MiT). Although there are only two limbs, MiT does not produce satisfactory results, since either one has risks to lose track and there are no other constraints to get it back except image observations, and MiT hardly produce plausible results satisfying the physical link constraints. Some frames of MiT are shown in Fig. 3.9.

The second test sequence is on a 3-part finger and consists of 182 frames. As expected, MFMC produces very robust and stable results. Sample result images are shown in Fig. 3.10.

The third test sequence is on a 6-part upper body, in which complex arm motions as well as global movement of the torso and head are presented. The sequence consists of 834 frames.

[1]All video results are available upon request

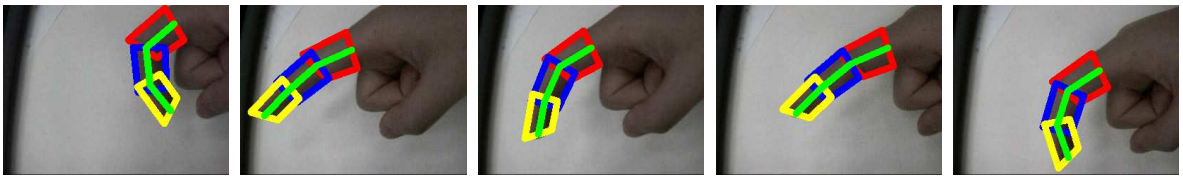


Figure 3.10. Mean field Monte Carlo (MFMC): tracking 3-part finger.

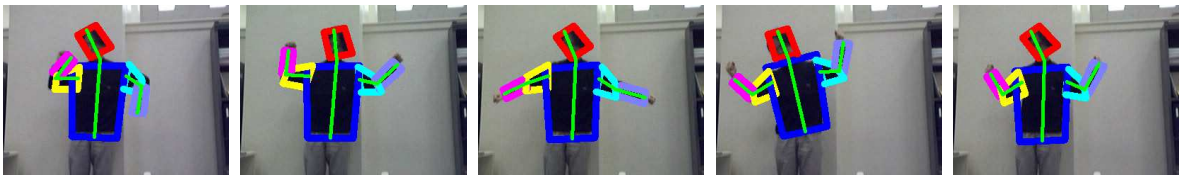


Figure 3.11. Mean field Monte Carlo (MFMC): tracking 6-part upper body.

Although the articulation is quite complicated, it does not fail the MFMC algorithm. Sample result images are shown in Fig. 3.11.

The most complicated test sequence we have experimented is on 10-part full body motion, and the sequence has 767 frames. Arms and legs are the most articulated body parts, and they present significant motion. None of our run of MiT succeed, because a leg is easy to get lost and never be able to come back. Sample result images of MiT are shown in Fig. 3.12. When MFMC is applied, the tracking result is still very stable unlike MiT. It is able to track the articulated motion until frame 368. Through subjective evaluation, the tracking quality does not decrease due to the increase of the complexity of the articulation. Sample result images are shown in Fig. 3.13. The MFMC algorithm runs on a single processor PC of 2.0GHz running Window XP. We do not perform any code optimization. For all these experiments, the number of mean field iterations is set to 5. The number of particles for each part and the frame rates are shown in Table 3.1. As we can clearly observe from the table, with 200 samples for each limb, the processing frame rates decrease almost linearly w.r.t.

experiments	2-part	3-part	6-part	10-part
particles/part	200	200	200	200
frame/second	2.02	1.28	0.94	0.56

Table 3.1. A comparison of the computation of different articulated objects. The exponential requirement for computation is overcome as expected.

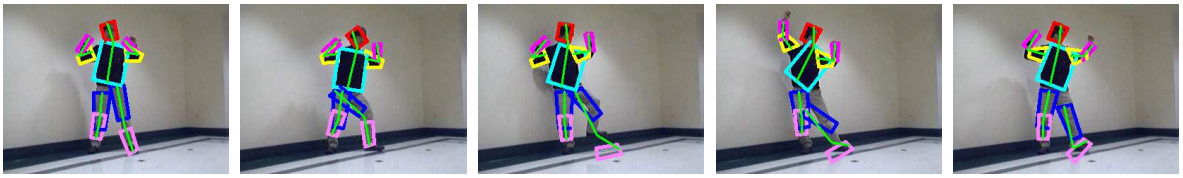


Figure 3.12. Multiple independent tracker (MiT): tracking 10-part full body.

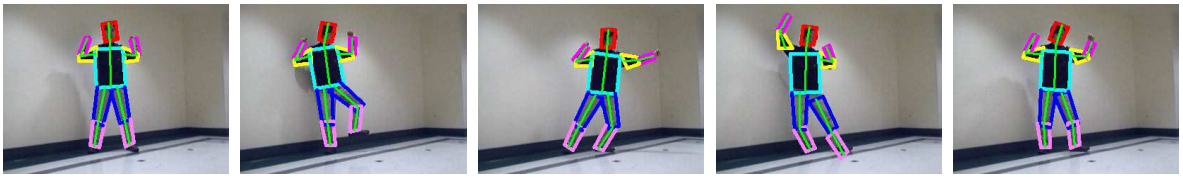


Figure 3.13. Mean field Monte Carlo (MFMC): tracking 10-part full body.

the number of limbs. This empirically demonstrates that the proposed MFMC algorithm does achieve linear complexity w.r.t. the number of limbs.

3.8. Conclusion

Tracking articulated objects is a challenging problem, since the increase of the number of limbs and the physical connection constraints of them would potentially incur high dimensionality, and fail tracking algorithms developed for single target. Thus, algorithms with close to linear complexity would have much better scalability. In this chapter, we propose a collaborative approach to achieve such a goal. Instead of using the joint angle representation which is irreducible, we adopt a highly redundant representation for articulated body, i.e., represent individual limb by its own motion parameters, but reinforce the constraints

of different limbs by a Markov network. Variational analysis is performed for probabilistic reasoning on this graphical model. Interestingly, a set of fixed point equations (i.e., the mean field equations) is found, which suggests a collaborative solution to the problem through the iterative interactions among neighborhood limbs. Then a mean field Monte Carlo (MFMC) algorithm is designed to achieve effective computation. In the context of dynamics, we propose a dynamic Markov network model and MFMC is extended to a sequential MFMC algorithm for visual tracking. Extensive experiments clearly validate our approach.

One of the future work is to extend the algorithm to 3D. Since self-occlusion seems a severe issue for articulated motion, another possible direction is to design collaborative algorithms for solving the occlusion problem. Moreover, since a centralized joint angle representation may facilitate the incorporation of high-order motion constraints, another possible future work would be to design algorithms that combine centralized and decentralized representation together to achieve more efficient and more accurate tracking of the articulated body.

CHAPTER 4

Variational maximum a posterior estimation

As we have discussed in Chapter 3, the proposed collaborative approach reveals the probabilistic integration of three measurement sources: the local likelihood, the dynamic priors, and the neighborhood priors. One challenge behind such a integration process is that these visual measurements tend to have multiple modes. This directly results in the multimodality of the integrated results, i.e., there are several results obtained from the visual measurement integration, only one of them is true though. One approach to addressing this issue is to keep multiple hypothesis for the integration results. However, in many cases, a unique result to the target problem is necessary. Therefore, we must identify the optimal mode of the integrated visual measurements and regard it as the final result. This is what this chapter is about, i.e., an efficient probabilistic variational method to achieve that.

4.1. Introduction

Bayesian inference methods recover the posterior distribution $P(\mathbf{X}|\mathbf{Z})$, or find the maximum a posteriori (MAP) estimation $\hat{X} = \arg\max_X\{P(\mathbf{X}|\mathbf{Z})\}$, where $\mathbf{Z}$ is the set of observations of the stochastic system, and $\mathbf{X}$ is the underlying stochastic processes generating $\mathbf{Z}$. Many real problems can be effectively modeled and solved under the Bayesian inference framework. In the literature of signal processing and computer vision, Bayesian methods are widely used in signal estimation [**2**], image segmentation [**3**, **4**, **116**], image super-resolution [**29**, **30**] and visual tracking [**106**, **126**, **130**], etc.. Many of these Bayesian

inference problems are formulated and represented by probabilistic graphical models [**29**, **30**, **106**, **126**, **130**].

Most traditional methods of Bayesian inference such as the belief propagation (BP) algorithm [**29**, **30**, **134**] and the variational inference methods [**6**, **46**, **123**, **126**] focus on recovering either the exact or the approximate posterior distributions. The problem is that even if we could obtain the exact posterior distribution, in general it is still very difficult to find the MAP estimate since it involves global optimization. The Markov chain Monte Carlo (MCMC) technique with simulated annealing (SA) [**33**, **65**, **79**] provides a principled way to search for the global optimum of the posterior and the convergence in probability to the global optimum has been proven [**33**]. However, the SA schemes are usually computationally intensive, which hinders their applicability in many real applications.

In this chapter, we propose an efficient approach to finding the MAP estimate by an annealed mean field variational analysis. We show that when the covariance of a variational Gaussian distribution approaches to zero, the infimum point of the KL divergence between the variational Gaussian and the real posterior will be the same as the supreme point of the real posterior. Thus in the limit, minimizing the KL divergence between the variational Gaussian and the real posterior is equivalent to maximizing the real posterior. The advantage of minimizing the former is that we can nicely incorporate a deterministic annealing (DA) scheme [**70**, **88**, **89**, **138**] into the mean field fixed-point iterations, which will eventually converge into the optimal or a near-optimal maximum point of the real posterior. This new method, namely variational MAP, is an efficient and effective method for obtaining the MAP estimate of a complex stochastic system.

The remainder of this chapter is organized as follows. In Sec. 4.2, related work are categorized and discussed. Then in Sec. 4.3, we construct the theoretic foundation of the

variational MAP algorithm by revealing a general theorem of the KL divergence between a Gaussian and an arbitrary p.d.f.. In Sec. 4.4, without loss of generality, we deduce the mean field fixed-point iterations under a Markov network, where the mean field approximation is constrained to be a multivariate Gaussian. We then propose the variational MAP algorithm in Sec. 4.5. Furthermore, a Monte Carlo implementation of such a variational MAP algorithm is proposed in Sec. 4.6. Extensive experimental results are demonstrated and discussed in Sec. 4.7. Finally we conclude our work and propose the possible future work in Sec. 4.8.

4.2. Related work

We propose the variational MAP algorithm under the context of graphical models, since it is a powerful means of representing real stochastic systems. Moreover, the MAP estimate involves global optimization. Related work can thus be categorized into three. The first category is related to graphical model representation of stochastic systems. The second category involves the Bayesian inference algorithms on graphical models. While the third category is related to the global optimization methods.

Bayesian network (BN), dynamic Bayesian network (DBN) [**83**,**85**], Markov network [**29**, **30**] and dynamic Markov network [**41**, **46**, **126**] are all typical graphical models [**58**]. They are widely used for modeling and solving computer vision problems. To mention some, a BN is proposed in [**120**] for spatial-temporal segmentation of video sequences. Various DBNs are proposed to address different problems in visual tracking, such as multiple cue co-inference [**130**], switching observation models for contour tracking in clutter [**125**] and tracking the appearances of multiple targets against occlusion [**132**]. The Markov network is adopted to achieve image super-resolution [**29**, **30**]. While various dynamic Markov networks are adopted to perform articulated human body tracking [**126**], to analyze structured

deformable shapes [**46**] and to formulate a rigorous bi-directional multi-scale visual tracking algorithm to address the abrupt motion [**41**]. Although there are many types of graphical models, they all can be transformed into one another [**58**].

For Bayesian inference in graphical models, when there is no loop, the sum-product algorithm or belief propagation (BP) [**29**, **58**] can obtain the exact inference efficiently through a local message passing process. When there are loops, the loopy BP [**82**] and generalized BP [**134**] can obtain good approximate results [**29**, **31**]. As an approximation, Monte Carlo techniques such as Markov chain Monte Carlo (MCMC) [**3**, **4**, **58**, **116**] and sequential Monte Carlo [**9**, **52**, **53**] can be used for implementing the Bayesian inference by sampling. In addition, probabilistic variational approach provides a principled way for approximate inference such as the mean field variational analysis [**6**, **46**, **54**, **123**, **126**], which seeks the best approximate results by minimizing the KL divergence between the mean field distribution and the real posterior distribution.

The non-parametric BP [**110**] and the PAMPAS algorithm [**51**] are proposed to implement the Bayesian inference on complex real valued graphical models by combining BP with a MCMC sampler. A different approach is the sequential mean field Monte Carlo algorithm (SMFMC) [**46**, **126**], which combines the mean field variational analysis with the sequential Monte Carlo technique. It is also proposed to implement efficient Bayesian inference on complex real valued graphical models.

Finding the MAP estimate is a global optimization problem. In terms of complexity, it is a NP-hard problem in the combinatory context. However, the stochastic simulated annealing (SA) [**33**, **65**, **79**] can achieve good results in many applications since the convergence in probability to the global optimum is proven [**33**]. But SA algorithms are often inherently slow due to their randomized local search strategy. Deterministic annealing (DA) [**89**, **138**]

methods intend to overcome the inefficiency of the SA methods. They are based on deterministic optimization scheme, but they incorporate stochastic smoothing by optimizing over a probabilistic state space [**89**]. Although global optimality may not be guaranteed for DA, many empirical studies have shown that the DA methods are very likely to achieve optimal or near optimal solutions [**89**]. The annealing methods are enlightened by the annealing process of a thermodynamic system, which drives the system to stay in the lowest energy and thus most probable state. Annealing methods have been widely used in image processing, computer vision and pattern recognition for robust M-Estimation [**70**], for designing piecewise regression models [**96**], for image texture segmentation and grouping [**88**] and for object recognition [**23**], to list a few.

In [**22**], an annealed particle filtering algorithm, which integrates a SA scheme with the sequential Monte Carlo algorithm, is proposed to find the maximum of the articulated human motion posteriors. Instead of using MCMC, weighted re-sampling is preformed during the SA process. Notwithstanding the empirically demonstrated effectiveness, this algorithm is largely based on heuristics and there is no strict theoretic proof about the convergence of such a process.

The variational MAP algorithm [**42**, **43**] proposed in this chapter integrates the mean field variational inference method [**6**, **46**, **58**, **123**, **126**] with a DA scheme [**88**, **89**, **138**]. By constraining the mean field variational distribution to be a multivariate Gaussian, the covariance of the Gaussian will be used as the "temperature" for annealing. In each step of the annealing, we iterate the Gaussian mean field fixed-point equations to convergence. As the covariance of the variational Gaussian approaches to zero, the mean of it will be very likely to converge into the global maximum point or a near global maximum point of the real posterior. Although the original mean field variational method [**58**, **126**] can only obtain an

approximation of the real posterior, the proposed variational MAP algorithm can find the exact optimal or near-optimal MAP estimate. It is an efficient and direct MAP inference algorithm for complex stochastic systems.

4.3. Kullback-Leibler divergence between a Gaussian and an arbitrary p.d.f.

The KL divergence or relative entropy between two probabilistic distribution $g(\mathbf{x})$ and $p(\mathbf{x})$ is defined as

$$KL\left(g(\mathbf{x})\|p(\mathbf{x})\right) = \int_{\mathbf{x}} g(\mathbf{x}) \log \frac{g(\mathbf{x})}{p(\mathbf{x})} d\mathbf{x}. \tag{4.1}$$

It is a measurement of the dissimilarity between two distributions. It has the property that it is zero if $g(\mathbf{x})$ and $p(\mathbf{x})$ are equal almost everywhere (a.e.) and positive otherwise. But it is not a real distance since it is not symmetric, i.e., $KL(g(\mathbf{x})\|p(\mathbf{x})) \neq KL(p(\mathbf{x})\|g(\mathbf{x}))$. Generally, minimizing $KL(g(\mathbf{x})\|p(\mathbf{x}))$ w.r.t. $g(\mathbf{x})$ will favor those $g(\mathbf{x})$ distributions whose probability densities all lie in the regions with high probability under $p(\mathbf{x})$, but without the requirement that all those areas are covered. While minimizing $KL(p(\mathbf{x})\|g(\mathbf{x}))$ w.r.t. $g(\mathbf{x})$ will favor the settings of $g(\mathbf{x})$ which can cover all the high probability areas in $p(\mathbf{x})$, even if this will result in assigning the high probability area of $g(\mathbf{x})$ to the very low probability area of $p(\mathbf{x})$ [**123**].

It is also worth noting that the KL divergence in Eq. 4.1 is finite only when $g(\mathbf{x})$ and $p(\mathbf{x})$ have the same support(we set $0 \log \frac{0}{0} = 0$, which is motivated by continuity.) [**20**]. Thus if $g(\mathbf{x})$ is a Gaussian and $p(\mathbf{x})$ is compactly supported, the $KL(g(\mathbf{x})\|p(\mathbf{x}))$ will be $+\infty$.

Based on the above observations, if we constrain the $g(\mathbf{x})$ distribution to be a Gaussian distribution, we have the following theorem relating the supreme of $p(\mathbf{x})$ and the infimum of $KL(g(\mathbf{x})\|p(\mathbf{x}))$. We must emphasize beforehand that the integrability assumption in

Eq. 4.2 is essential otherwise the $KL(g(\mathbf{x})\|p(\mathbf{x}))$ could be $+\infty$ no matter how the Gaussian distribution $g(\mathbf{x})$ is translated and scaled.

Theorem 4.3.1. *Let $p(\mathbf{x})$, $\mathbf{x}$ is a random vector in $\mathcal{R}^n$, be a bounded, continuous and everywhere positive p.d.f. with the properties:*

- *There exists a unique $\mathbf{x}^* \in \mathcal{R}^n$ such that $p(\mathbf{x}^*) = \sup_{\mathbf{x}\in\mathcal{R}^n} p(\mathbf{x})$*
- *$p(\mathbf{x})$ is proper, i.e., $p(\mathbf{x}) \to \mathbf{0}$ as $\mathbf{x} \to \infty$*
- *The following integrability condition in Eq. 4.2 holds*

$$\left| \int_{\mathbf{x}} exp\left\{ -\frac{\mathbf{x}^T\mathbf{x}}{2} \right\} \log p(\mathbf{x}) d\mathbf{x} \right| < +\infty \tag{4.2}$$

Suppose $q(\mathbf{x}) \sim \mathcal{N}(\mathbf{x}|\mathbf{0}, \mathcal{I}_n)$ is a Gaussian distribution with zero mean and identity covariance matrix $\mathcal{I}_n$, then denote $q_\sigma^{\tilde{\mu}}(\mathbf{x}) \sim \mathcal{N}(\mathbf{x}|\tilde{\mu}, \sigma^2\mathcal{I}_n), \mathbf{x} \in \mathcal{R}^n$ as the Gaussian distribution with mean $\tilde{\mu}$ and diagonal covariance $\sigma^2\mathcal{I}_n$. Assume $\tilde{\mu}_\sigma$ is such that $KL(q_\sigma^{\tilde{\mu}_\sigma}(\mathbf{x})\|p(\mathbf{x})) = \inf_{\tilde{\mu}} KL(q_\sigma^{\tilde{\mu}}(\mathbf{x})\|p(\mathbf{x}))$, then

$$\lim_{\sigma\to 0} \tilde{\mu}_\sigma = \mathbf{x}^*. \tag{4.3}$$

Proof. The proof can be found in Appendix B based on several Lemmas in Appendix A. □

Eq. 4.3 in Theorem 4.3.1 nicely reveals to us a DA scheme to find the maximum point of $p(\mathbf{x})$, i.e., we can minimize w.r.t. $\tilde{\mu}$ a series of $KL(q_\sigma^{\tilde{\mu}}(\mathbf{x})\|p(\mathbf{x}))$. This can be achieved by initially setting the σ^2 to be very large value and decreasing it asymptotically to zero. When the σ^2 is very large, the optimization of $KL(q_\sigma^{\tilde{\mu}}(\mathbf{x})\|p(\mathbf{x}))$ is just a convex optimization problem [**89**]. With the decreasing of the σ^2, the $KL(q_\sigma^{\tilde{\mu}}(\mathbf{x})\|p(\mathbf{x}))$ will have more local

minima and the optimization is more complex. For a fixed σ^2, we can run an optimization algorithm to find the minimum of $KL(q_\sigma^{\tilde{\mu}}(\mathbf{x})\|p(\mathbf{x}))$, then the result will be used as the initial point of the optimization in the next step of annealing. As σ^2 decreases asymptotically to zero, the whole annealed optimization process will be very likely to converge into the global minimum of the $KL(q_\sigma^{\tilde{\mu}}(\mathbf{x})\|p(\mathbf{x}))$, and thus the global maximum of $p(\mathbf{x})$.

Moreover, in many cases, the $p(\mathbf{x})$ is not directly in hand, so we may not be able to maximize it directly. For example, in the Bayesian inference problem presented in Sec. 4.4, where $p(\mathbf{x})$ is corresponding to the posterior distribution which must be inferred from the observations. In Sec. 4.5, we show that by using a novel variational inference framework, the problem of optimal MAP estimation can be efficiently solved by minimizing the KL divergence between a variational Gaussian and the real posterior distribution without explicitly recovering the latter.

4.4. Gaussian mean field variational analysis

In this section, we present the Gaussian constrained mean field variational analysis, which functions as the optimization method in one annealing step in the variational MAP algorithm. To better illustrate it, we adopt a specific type of graphical model, i.e., the Markov network as shown in Fig. 4.1. Since different types of graphical models can be transformed to one another [**58**], adopt a specific type of graphical model will not lose the generality of the proposed algorithm.

In a Markov network, each $\mathbf{z}_i$ represents an observation of the latent random variable $\mathbf{x}_i$. Each undirected link is associated with a potential function $\psi_{ij}(\mathbf{x}_i, \mathbf{x}_j)$, which models the probability of two adjacent nodes being in a certain state pair. Each directed link represents an observation function $\phi_i(\mathbf{z}_i|\mathbf{x}_i)$ which models the probability of the observation $\mathbf{z}_i$ given $\mathbf{x}_i$.

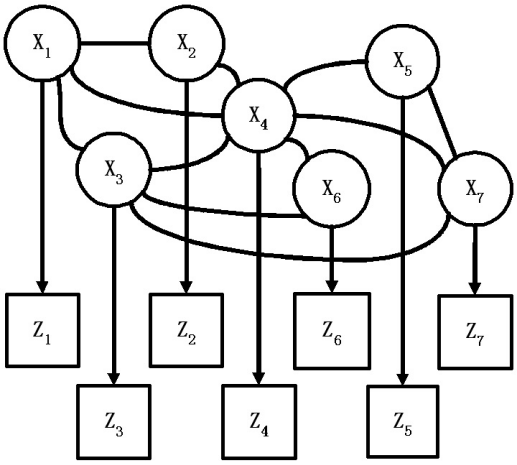


Figure 4.1. An example of Markov network.

Denotes $\mathbf{X} = \{\mathbf{x}_i, i = 1 \dots M\}$ as the set of latent random variables and $\mathbf{Z} = \{\mathbf{z}_i, i = 1 \dots M\}$ as the set of all observations. Then the joint probability of the Markov network is

$$P(\mathbf{X}, \mathbf{Z}) = \frac{1}{C} \prod_{\{i,j\} \in \mathcal{E}} \psi_{ij}(\mathbf{x}_i, \mathbf{x}_j) \prod_{i \in \mathcal{V}} \phi_i(\mathbf{z}_i | \mathbf{x}_i), \tag{4.4}$$

where $\mathcal{E}$ is the set of undirected links, $\mathcal{V}$ is the set of directed links, and C is a normalization constant. Note here we only define 2nd order potentials in the hidden layer for $\mathbf{X}$. Then the Bayesian MAP inference in the Markov network is to find

$$\hat{\mathbf{X}} = \arg\max_{\mathbf{X}} P(\mathbf{X}|\mathbf{Z}). \tag{4.5}$$

We show that by combining the mean field variational method [**6**, **45**, **46**, **54**, **123**, **126**] with a DA scheme [**89**, **138**], we can efficiently find the optimal or near optimal MAP estimation of the joint posterior $P(\mathbf{X}|\mathbf{Z})$.

To achieve that, firstly, we adopt the mean field approximation, i.e.,

$$P(\mathbf{X}|\mathbf{Z}) \approx Q(\mathbf{X}) = \prod_{i=1}^{M} Q_i(\mathbf{x}_i). \tag{4.6}$$

Suppose all the random variables share one common dimension N, we further constrain each of the $Q_i(\mathbf{x}_i)$ as a multivariate Gaussian, i.e.,

$$Q_i(\mathbf{x}_i) \sim \mathcal{N}(\mathbf{x}_i|\tilde{\mu}_i, \mathbf{\Sigma}_i), \tag{4.7}$$

where $\tilde{\mu}_i$ is the N dimensional mean vector and $\mathbf{\Sigma}_i = \sigma^2\mathcal{I}_N$ is a fixed $N \times N$ diagonal covariance matrix. Then $Q(\mathbf{X})$ is a $N \times M$ dimensional multivariate Gaussian distribution with $NN \times NM$ dimensional diagonal covariance matrix as follows:

$$Q(\mathbf{X}) \sim \mathcal{N}(\mathbf{X}|\tilde{\mu}, \mathbf{\Sigma}) = \mathcal{N}\left(\mathbf{X} \middle| \begin{bmatrix} \tilde{\mu}_1 \\ \tilde{\mu}_2 \\ \cdot \\ \cdot \\ \tilde{\mu}_M \end{bmatrix}, \begin{bmatrix} \sigma^2\mathcal{I}_N & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \sigma^2\mathcal{I}_N & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdot & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdot & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \sigma^2\mathcal{I}_N \end{bmatrix}\right). \tag{4.8}$$

Following the general idea of variational inference, again, we can construct a cost function $J(Q)$ by Eq. 3.5. Now the constrained optimization problem is

$$\boxed{\begin{array}{l} \text{Maximize } J(Q) \\ \quad s.t. \quad Q_i(\mathbf{x}_i) \in \left\{\mathcal{N}(\mathbf{x}_i|\tilde{\mu}_i, \sigma^2\mathcal{I}_N)|\tilde{\mu}_i \in \mathcal{R}^N\right\}, \quad for \quad i = 1 \ldots M \end{array}} \tag{4.9}$$

We solve this constrained optimization problem by taking a strategy similar to the gradient projection method [**98**, **99**]. Firstly, we relax the constraint by letting $Q_i(\mathbf{x}_i)$ be any valid probabilistic distributions. Then we can perform the standard mean field variational

analysis presented in Sec. 3.4 to obtain the mean field fixed point equations [**45**, **46**, **54**, **126**], i.e., for each $i = 1 \ldots M$, we have

$$Q_i(\mathbf{x}_i) = \frac{1}{C_i} e^{E_Q[\log P(\mathbf{X},\mathbf{Z})|\mathbf{x}_i]}, \tag{4.10}$$

where C_i is the normalization constant to assure that $Q_i(\mathbf{x}_i)$ be a valid probability density function. We can iterate this set of fixed-point equations in order to find a minimum point of $KL(Q(\mathbf{X})\|P(\mathbf{X}|\mathbf{Z}))$ when $Q(\mathbf{X})$ is a product of M Gaussian distributions with fixed covariance $\sigma^2\mathcal{I}_n$, i.e.,

$$\tilde{\mu}_i = \int_{\mathbf{x}_i} \mathbf{x}_i Q_i(\mathbf{x}_i) d\mathbf{x}_i \tag{4.11}$$

$$= \frac{1}{C_i} \int_{\mathbf{x}_i} \mathbf{x}_i e^{E_Q[\log P(\mathbf{X},\mathbf{Z})|\mathbf{x}_i]} d\mathbf{x}_i. \tag{4.12}$$

We call this set of fixed-point equations as *Gaussian mean field fixed point equations.* In fact, it is easy to figure out that Eq. 4.12 will minimize $KL(Q_i(\mathbf{x}_i)||\mathcal{N}(\mathbf{x}_i|\tilde{\mu}, \sigma^{\mathbf{2}}\mathcal{I}_{\mathbf{n}}))$ w.r.t. $\mathcal{N}(\mathbf{x}_i|\tilde{\mu}, \sigma^{\mathbf{2}}\mathcal{I}_{\mathbf{n}})$, where $Q_i(\mathbf{x}_i)$ is the unconstrained variational p.d.f. from Eq. 4.10, and $\mathcal{N}(\mathbf{x}_i|\tilde{\mu}, \sigma^{\mathbf{2}}\mathcal{I}_{\mathbf{n}})$ is a Gaussian distribution with fixed covariance $\sigma^2\mathcal{I}_n$. In this sense, Eq. 4.12 represents a projection of any p.d.f. $Q_i(\mathbf{x}_i)$ to the functional space spanned by all the Gaussian distributions with the fixed covariance $\sigma^2\mathcal{I}_n$. Then the projected Gaussian distribution will be used for the next mean field iteration. This process will continue until the mean field iterations reach the fixed-point. It exactly follows the same strategy of the gradient projection method [**98**, **99**].

Embedding Eq. 4.4 and Eq. 4.7 into Eq. 4.12, we obtain the set of factorized fixed-point equations, i.e.,

$$\tilde{\mu}_i = \frac{1}{C_i'} \int_{\mathbf{x}_i} \mathbf{x}_i \phi_i(\mathbf{z}_i|\mathbf{x}_i) e^{\sum_{j\in\mathcal{N}(i)} \int_{\mathbf{x}_j} \mathcal{N}(\mathbf{x}_j|\tilde{\mu}_j,\sigma^2\mathcal{I}_N) \log \psi_{ij}(\mathbf{x}_i,\mathbf{x}_j) d\mathbf{x}_j} d\mathbf{x}_i, \tag{4.13}$$

where C_i' is again a normalization constant and $\mathcal{N}(i)$ indicates the set of neighboring nodes of $\mathbf{x}_i$. Then we iteratively assign $Q_i(\mathbf{x}_i) = \mathcal{N}(\mathbf{x}_i|\tilde{\mu}_i, \sigma^2\mathcal{I}_N)$ where $\tilde{\mu}_i$ is calculated according to Eq. 4.13. Please note that the covariance of the variational Gaussian distribution is kept fixed during the fixed-point iteration and projection process.

For a constant $\mathbf{\Sigma} = \sigma^2\mathcal{I}_{NM}$, Eq. 4.13 is the Gaussian mean field fixed-point equation to update $\tilde{\mu}_i$. We can iterate this set of fixed-point equations, and $\tilde{\mu}_i$ will converge to a minimum point of $KL(Q(\mathbf{X})||P(\mathbf{X}|\mathbf{Z}))$. This set of fixed-point equations is efficient since the updating of each $\tilde{\mu}_i$ only involves the local computation in the neighborhood of $\mathbf{x}_i$ in the graphical model.

However, another issue of interest is that to solve the constrained maximization of $J(Q)$, we may directly take the derivative of $J(Q)$ w.r.t. the mean $\tilde{\mu}_i$ of each of the Gaussian $Q_i(\mathbf{x}_i)$ and set them to zero. By interchanging the derivative and integral in Eq. 3.8 we can then obtain the following equations

$$\tilde{\mu}_i = \frac{\int_{\mathbf{x}_i} \mathbf{x}_i Q_i(\mathbf{x}_i) E_Q\{\log P(\mathbf{X},\mathbf{Z})|\mathbf{x}_i\} d\mathbf{x}_i}{\int_{\mathbf{x}_i} Q_i(\mathbf{x}_i) E_Q\{\log P(\mathbf{X},\mathbf{Z})|\mathbf{x}_i\} d\mathbf{x}_i}. \tag{4.14}$$

Again, by embedding Eq. 4.4 into Eq. 4.14, we obtain the factorized version of Eq. 4.15, i.e.,

$$\tilde{\mu}_i = \frac{1}{C_i''} \int_{\mathbf{X}} \mathbf{x}_i \prod_{j\in\mathcal{V}} Q_j(\mathbf{x}_j) \left(\sum_{(k,l)\in\mathcal{E}} \log \psi_{kl}(\mathbf{x}_k,\mathbf{x}_l) + \sum_{m\in\mathcal{V}} \log \phi_m(\mathbf{x}_m) \right) d\mathbf{X}, \tag{4.15}$$

where C_i'' is again a normalization constant.

While it seems that Eq. 4.15 be a more direct solution, our experiments show that even in a relative simple synthetic problem as that in Sec. 4.7.2, the iteration of Eq. 4.15 failed to converge. Two reasons might explain why this happens: (1). the deduction of Eq. 4.14 involves an interchange between derivative and integral, which may not be justified; (2). the iteration of Eq. 4.15 is not numerically stable, i.e., it might be easily got trapped in some saddle points. Another reason that we adopt Eq. 4.11 is that the updating of $\tilde{\mu}_i$ only involves the local computation in the neighborhood of the node $\mathbf{x}_i$, while Eq. 4.15 does not have such kind of nice local property. Therefore, Eq. 4.11 is more justified as well as more computationally efficient than Eq. 4.15.

4.5. Variational MAP by deterministic annealing

Based on Theorem 4.3.1 in Sec. 4.3 and the multivariate Gaussian constrained mean field variational analysis in Sec. 4.4, we show that we can nicely adopt a DA scheme to efficiently find the optimal MAP estimate without explicitly recovering the $P(\mathbf{X}|\mathbf{Z})$.

We firstly relax the problem of estimating the global maximum of $P(\mathbf{X}|\mathbf{Z})$, i.e., we can instead minimize $KL(Q(\mathbf{X})\|P(\mathbf{X}|\mathbf{Z}))$ where $Q(\mathbf{X})$ is constrained to be a multivariate Gaussian with a fixed diagonal covariance $\mathbf{\Sigma} = \sigma^2\mathcal{I}_{NM}$ as in Eq. 4.8. We can then apply the DA scheme revealed by Theorem 4.3.1. This is achieved by regarding the σ^2 as the temperature T for annealing. We can set it to be very large at the start. The minimization of the $KL(Q(\mathbf{X})\|P(\mathbf{X}|\mathbf{Z}))$ in this start setting is usually a trivial convex optimization problem [**89**]. Then the multivariate Gaussian constrained mean field iteration in Eq. 4.13 can usually find the only minimum point under this setting. Using this result as an initialization, we decrease σ^2 to be smaller toward zero and run the mean field iteration in Eq. 4.13

Variational Maximum a Posteriori Algorithm

(1) `Initialization:` Annealing control parameter $m = 0$; T_{max} be very large where the annealing starts and T_{min} be very small near zero where the annealing stops; let $\mathcal{I}_N$ be the $N \times N$ identity matrix; denote $\vec{\mu}_{i,0}$ as the set of randomly generated mean vector for initialization.

(2) `Annealing:` $m = m + 1$, $T = \frac{T_{max}}{m}$, then $\mathbf{\Sigma}_i = T \times \mathcal{I}_N$; $\vec{\mu}_{i,m} = \vec{\mu}_{i,m-1}$, $i = 1 \dots M$; If $T > T_{min}$, goto Step 3, else goto Step 4.

(3) `Mean field iteration:` Update $\vec{\mu}_{i,m}$ based on the current value of $\vec{\mu}_{i,m}$ and the fixed $\mathbf{\Sigma}_i$ according to Eq. 4.13. Iterate this step until convergence. Then jump back to Step 2.

(4) `Result:` $\vec{\mu}_i^* = \vec{\mu}_{i,m}$, $i = 1 \dots M$, are the MAP estimation of $P(\mathbf{X}|\mathbf{Z})$.

Figure 4.2. Variational MAP algorithm.

again. We can repeat the process until the σ^2 decreasing to near zero. Then upon convergence, the whole annealing process will be very likely to obtain the global minimum of the $\lim_{\sigma \to 0} KL(Q(\mathbf{X}) \| P(\mathbf{X}|\mathbf{Z}))$, and thus the global maximum of $P(\mathbf{X}|\mathbf{Z})$. Therefore, we only need to control one parameter $T = \sigma^2$ for the annealing process. Generally, we propose the variational MAP algorithm as shown in Fig. 4.2.

Nevertheless, the annealing scheme, i.e., the decreasing rate of T, needs to be carefully designed to have a good optimization result. Unfortunately, it seems that a theoretic analysis of the annealing rate is very difficult. In the proposed algorithm, we let the annealing control parameter T decrease hyperbolically with the annealing number K. In our experiments, such an annealing scheme always obtains satisfactory results. Please note that although the mean field variational analysis can only obtain an approximate posterior, the proposed algorithm is very likely to obtain the exact optimal MAP estimate.

4.6. Monte Carlo simulation of the variational MAP

In a real valued graphical model such as that in Fig. 4.1, if all the observation functions $\phi_i(\mathbf{z}_i|\mathbf{x}_i)$ and all the potential functions $\psi_{ij}(\mathbf{x}_i, \mathbf{x}_j)$ are Gaussian, then we may obtain a closed

form analytical solution of the fixed-point equations in Eq. 4.13. However, either $\phi_i(\mathbf{z}_i|\mathbf{x}_i)$ or $\psi_{ij}(\mathbf{x}_i, \mathbf{x}_j)$ can be complex non-Gaussian distributions, e.g., the image observation function in the CONDENSATION contour tracker [**9**, **52**, **53**] is the interference of a Gaussian random process and a Poisson random process due to the background clutter. This makes it very difficult to obtain analytical solutions to the fixed-point equations in Eq. 4.13, e.g., it would be very difficult to evaluate the normalization constant C_i', since it involves multiple integrals of complex distributions.

Nevertheless, under the non-Gaussian case, we can seek the help of Monte Carlo simulation to approximately evaluate Eq. 4.13. According to the strong law of large numbers, as the number of *i.i.d.* samples from a distribution approaches to infinity, any order of the sample quadrature will converge to the same order of distribution statistics with probability one. Thus, to evaluate Eq. 4.13, firstly, we can generate M sets of samples to approximate each of the $Q_i(\mathbf{x}_i)$, i.e.,

$$Q_i(\mathbf{x}_i) = \mathcal{N}(\mathbf{x}_i|\vec{\mu}_i, \sigma^2\mathcal{I}_N) \sim \{\mathbf{s}_{i,k}\}_{k=1}^{K}, i = 1 \ldots M, \tag{4.16}$$

where K is the number of samples used for simulation. Then these M sets of samples can be used for evaluating Eq. 4.13 approximately, i.e.,

$$\vec{\mu}_i = \frac{1}{C_i'''} \sum_{k=1}^{K} \mathbf{s}_{i,k} \phi_i(\mathbf{z}_i|\mathbf{s}_{i,k}) \exp\left(\sum_{j \in \mathcal{N}(i)} \frac{1}{K} \sum_{l=1}^{K} \log \psi_{ij}(\mathbf{s}_{i,k}, \mathbf{s}_{j,l}) \right), \tag{4.17}$$

where

$$C_i''' = \sum_{k=1}^{K} \phi_i(\mathbf{z}_i|\mathbf{s}_{i,k}) \exp\left(\sum_{j \in \mathcal{N}(i)} \frac{1}{K} \sum_{l=1}^{K} \log \psi_{ij}(\mathbf{s}_{i,k}, \mathbf{s}_{j,l}) \right). \tag{4.18}$$

Variational MAP Monte Carlo

(1) `Initialization:` Set the annealing control parameter $m = 0$; let T_{max} be very large where the annealing starts and T_{min} be very small near zero where the annealing stops; let $\mathcal{I}_N$ be the $N \times N$ identity matrix; denote $\vec{\mu}_{i,0}$ as the set of randomly generated mean vector for initialization.

(2) `Annealing:` $m = m + 1$, $T = \frac{T_{max}}{m}$, then $\mathbf{\Sigma}_i = T \times \mathcal{I}_N$; $\vec{\mu}_{i,m} = \vec{\mu}_{i,m-1}$, $i = 1 \ldots M$. If $T > T_{min}$, goto Step 3, else goto Step 4.

(3) `Mean field iteration:` Sample $\{\mathbf{s}_{i,k}\}_{k=1}^{K}$ from $Q_i(\mathbf{x}_i) = \mathcal{N}(\mathbf{x}_i|\vec{\mu}_{i,m}, \mathbf{\Sigma}_i)$ for $i = 1 \ldots M$, and calculate the updated $\vec{\mu}_{i,m}$ based on these sets of samples according to Eq. 4.17. Iterate this step until convergence. Then jump back to Step 2.

(4) `Result:` $\vec{\mu}_i^* = \vec{\mu}_{i,m}$, $i = 1 \ldots M$, are the MAP estimation of $P(\mathbf{X}|\mathbf{Z})$.

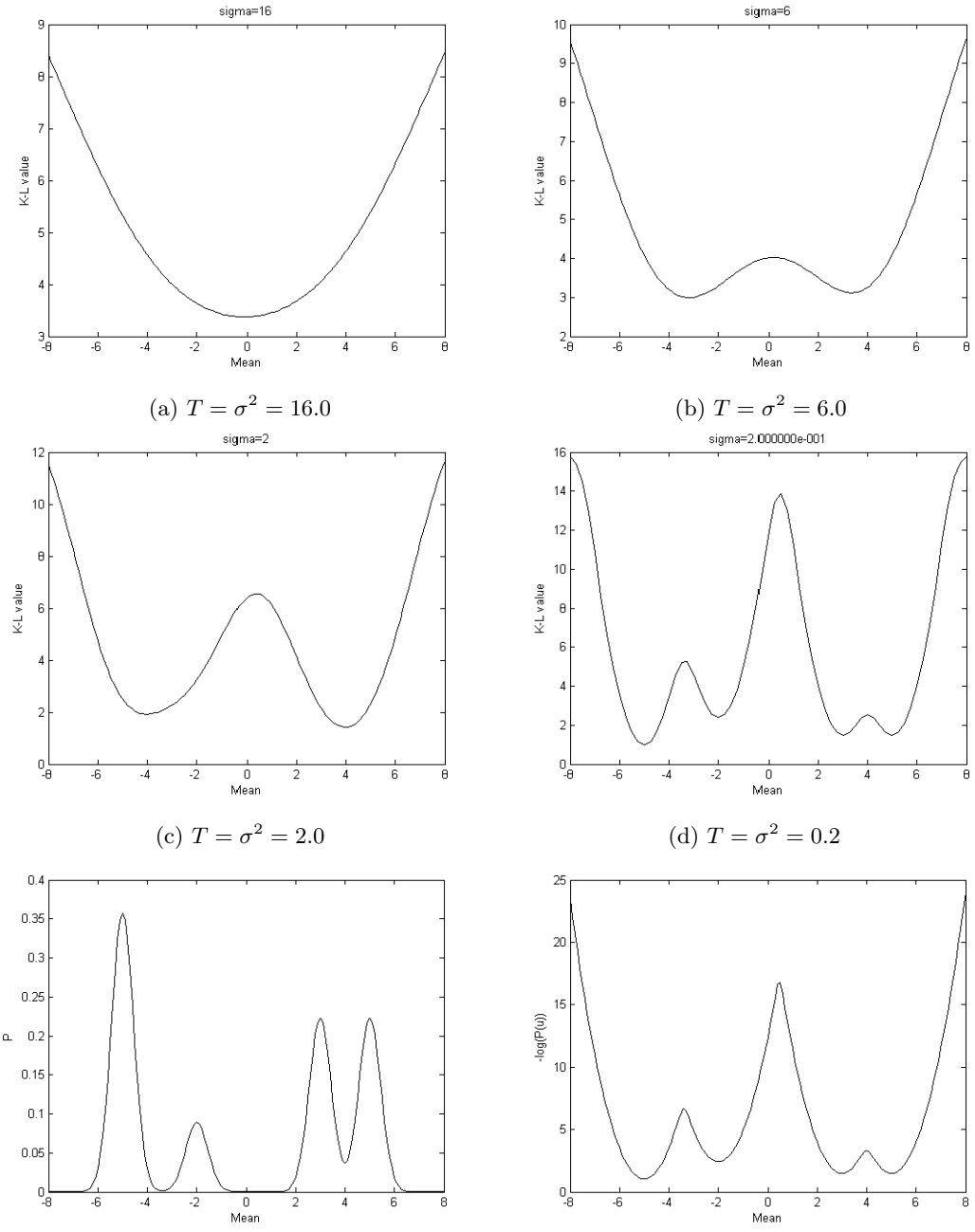
Figure 4.3. Monte Carlo implementation of the variational MAP algorithm.

is the normalization constant. Therefore, we propose the Monte Carlo implementation of the variational MAP algorithm in Fig. 4.3.

In fact, the use of Monte Carlo simulation in the variational MAP algorithm has other advantages in some computer vision applications. For example, in visual tracking, since the detection of the target is in general very difficult, it would be hard to obtain the image observation $\mathbf{z}_i$ and thus it is hard to evaluate the observation likelihood $\phi_i(\mathbf{z}_i|\mathbf{x}_i)$. Whereas in a sample based Monte Carlo algorithm, the observation likelihood $\phi(\mathbf{z}_i|\mathbf{x}_i)$ can be evaluated in a top-down approach, i.e., for each sample $\mathbf{s}_{i,k}$, we can easily match the model represented by the sample with the image data or image features corresponding to the sample, just as in the CONDENSATION contour tracker [**9**, **52**, **53**].

4.7. Validation experiments and application to articulated body tracking

In this section, we present extensive experimental results of both synthetic problems and real applications, which demonstrate the effectiveness and efficiency of the proposed variational MAP algorithm.



(a) $T = \sigma^2 = 16.0$

(b) $T = \sigma^2 = 6.0$

(c) $T = \sigma^2 = 2.0$

(d) $T = \sigma^2 = 0.2$

(e) $p(\mu)$

(f) $-log(p(\mu))$

Figure 4.4. Evolution of the $KL(q(\mathbf{x})\|p(\mathbf{x}))$ w.r.t. μ during annealing: (a). the KL topology when $T = 16.0$; (b). the KL topology when $T = 6.0$; (c). the KL topology when $T = 2.0$; (d). the KL topology when $T = 0.2$; (e). the plot of the Gaussian mixture $p(\mu)$. (f) the plot of the $-\log p(\mu)$.

4.7.1. Evolution of the topology of the KL divergence during annealing

In this experiment, we use an illustrative example to present the topology of the KL divergence between a Gaussian distribution and a multi-modal Gaussian mixture during the process of annealing. As shown in Fig. 4.4, it does evolve as we expected from Theorem 4.3.1.

The real distribution $p(\mathbf{x}) = 0.4 \cdot \mathcal{N}(\mathbf{x}|-5, 0.2) + 0.1 \cdot \mathcal{N}(\mathbf{x}|-2, 0.2) + 0.25 \cdot \mathcal{N}(\mathbf{x}|3, 0.2) + 0.25 \cdot \mathcal{N}(\mathbf{x}|5, 0.2)$ is a Gaussian mixture of four kernels. The $q(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\mu, \sigma^2)$ is a Gaussian distribution. The annealing parameter is $T = \sigma^2$. We can observe in Fig. 4.4(a) that when T is large, i.e., $T = 16.0$, the $KL(q(\mathbf{x})\|p(\mathbf{x}))$ is really a convex function w.r.t. μ. Then, with the decreasing of T, the $KL(q(\mathbf{x})\|p(\mathbf{x}))$ will have more local minima, i.e., when $T = 6.0$ or $T = 2.0$, the $KL(q(\mathbf{x})\|p(\mathbf{x}))$ has two local minima as shown in Fig. 4.4(b) and Fig. 4.4(c). As the T decreases to near zero, i.e., $T = 0.2$, the $KL(q(\mathbf{x})\|p(\mathbf{x}))$ has four local minima at $\mu = -5.0, -2.0, 3.0, 5.0$, respectively. Each of them corresponds to one of the four local maxima of $p(\mathbf{x})$. The global minimum of $KL(q(\mathbf{x})\|p(\mathbf{x}))$ is at $\mu = -5.0$, which exactly corresponds to the global maximum of $p(\mathbf{x})$ at $\mathbf{x} = -5.0$, as shown in Fig. 4.4(d).

For comparison, we also present the plot of $p(\mu)$ in Fig. 4.4(e) and $-\log p(\mu)$ in Fig. 4.4(f). Compare Fig. 4.4(d) with Fig. 4.4(f), we empirically demonstrate that as a function of μ, the topology of $KL(q(\mathbf{x})\|p(\mathbf{x}))$ does converge to the topology $-\log p(\mu)$ as σ^2 approaches to zero. This result is what we expect from the Lemma 2 in the appendix A.

4.7.2. Variational MAP inference in an illustrative synthetic problem

To investigate the convergence of the proposed variational MAP algorithm, we perform it on an illustrative synthetic problem, which is modeled as a two-nodes Markov network in Fig. 4.5.

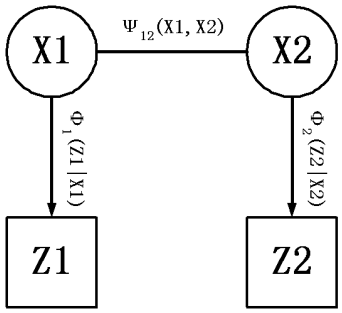


Figure 4.5. Two-nodes Markov network for the illustrative synthetic problem, where $\psi_{12}(\mathbf{x}_1, \mathbf{x}_2) = \mathcal{N}(\mathbf{x}_2 - \mathbf{x}_1|6.0, 0.3)$, $\phi_1(\mathbf{z}_1|\mathbf{x}_1) = 0.5\mathcal{N}(\mathbf{z}_1|\mathbf{x}_1 - 3.0, 0.3) + 0.4\mathcal{N}(\mathbf{z}_1|\mathbf{x}_1, 0.2) + 0.1\mathcal{N}(\mathbf{z}_1|\mathbf{x}_1 + 4, 0.4)$ and $\phi_2(\mathbf{z}_2|\mathbf{x}_2) = 0.3\mathcal{N}(\mathbf{z}_2|\mathbf{x}_2 - 5.0, 0.2) + 0.1\mathcal{N}(\mathbf{z}_2|\mathbf{x}_2 - 2.0, 0.3) + 0.4\mathcal{N}(\mathbf{z}_2|\mathbf{x}_2 + 3.0, 0.2) + 0.2\mathcal{N}(\mathbf{z}_2|\mathbf{x}_2 + 5.0, 0.1)$.

In this synthetic problem, both $\mathbf{x}_1$ and $\mathbf{x}_2$ are one dimensional random variables. The potential function between these two random variables is modeled as a Gaussian distribution , i.e.,

$$\psi_{12}(\mathbf{x}_1, \mathbf{x}_2) = \mathcal{N}(\mathbf{x}_2 - \mathbf{x}_1|6.0, 0.3) \tag{4.19}$$

The observation function $\phi_i(\mathbf{z}_i|\mathbf{x}_i)$, $i = 1, 2$ are modeled as two Gaussian mixtures respectively, i.e.,

$$\begin{aligned} \phi_1(\mathbf{z}_1|\mathbf{x}_1) &= 0.5\mathcal{N}(\mathbf{z}_1 - \mathbf{x}_1| - 3.0, 0.3) + 0.4\mathcal{N}(\mathbf{z}_1 - \mathbf{x}_1|0, 0.2) \qquad (4.20) \\ &+ 0.1\mathcal{N}(\mathbf{z}_1 - \mathbf{x}_1|4, 0.4) \\ \phi_2(\mathbf{z}_2|\mathbf{x}_2) &= 0.3\mathcal{N}(\mathbf{z}_2 - \mathbf{x}_2| - 5.0, 0.2) + 0.1\mathcal{N}(\mathbf{z}_2 - \mathbf{x}_2| - 2.0, 0.3) \qquad (4.21) \\ &+ 0.4\mathcal{N}(\mathbf{z}_2 - \mathbf{x}_2|3.0, 0.2) + 0.2\mathcal{N}(\mathbf{z}_2 - \mathbf{x}_2|5.0, 0.1). \end{aligned}$$

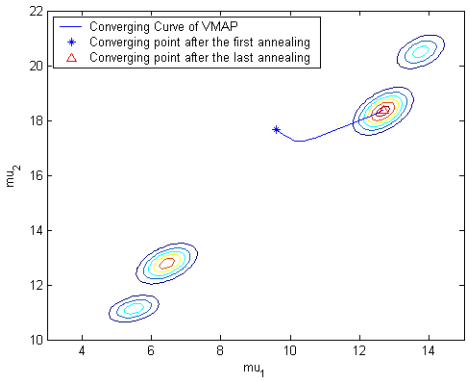
Then we randomly choose the observations $\mathbf{z}_1$ and $\mathbf{z}_2$ and perform the proposed variational MAP algorithm on it, we show the Bayesian MAP inference results in Fig. 4.6.

From Fig. 4.6(a), we can observe the convergence of the proposed variational MAP algorithm in this illustrative synthetic problem when $\{\mathbf{z}_1, \mathbf{z}_2\} = \{10.0, 16.0\}$. We randomly choose the initialization of μ_1 and μ_2 and run the algorithm many times, every time we obtain the same convergence curve, i.e., the converged result after the first step of annealing will always be the '$*$' shown in Fig. 4.6(a) at $\{\mu_1, \mu_2\} = \{9.6011, 17.6728\}$. This is what we expected since when T is very large, the $KL(\cdot)$ is a convex function and thus the optimization in this case will surely converge into the only minimum point, e.g., $\{\mu_1, \mu_2\} = \{9.6011, 17.6728\}$ in this case.

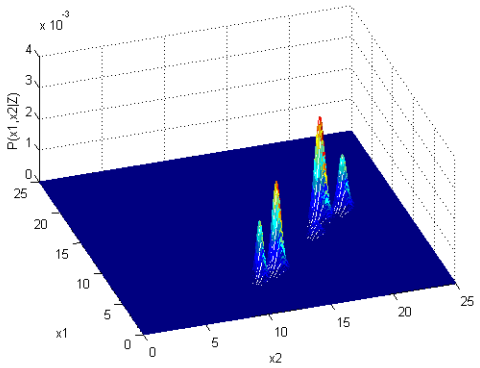
We can also observe that the proposed algorithm does converge to the global maximum of the posterior distribution, i.e., our algorithm converges at $\{\mu_1, \mu_2\} = \{12.6824, 18.3793\}$ which is shown as the '$\triangle$' in Fig. 4.6(a) and the numerically calculated MAP estimate is at around $\{\mathbf{x}_1, \mathbf{x}_2\} = \{12.70, 18.40\}$. Considering the possible error of the numerically calculated MAP estimate, we conclude that our algorithm does recover the global maximum of the posterior distribution $P(\mathbf{x}_1, \mathbf{x}_2|\mathbf{z}_1 = 10.0, \mathbf{z}_2 = 16.0)$. For comparison and visualization, we also present the topology of the posterior distribution $P(\mathbf{x}_1, \mathbf{x}_2|\mathbf{z}_1 = 10.0, \mathbf{z}_2 = 16.0)$ in Fig. 4.6(b).

Although in theory, we can not guarantee the algorithm to obtain the global optimal MAP estimation, extensive running of the experiments on the synthetic problem shows that the proposed variational MAP algorithm does always converge to the global maximum of the posteriori distribution. We present two other experimental results from Fig. 4.6(c) to Fig. 4.6(f). Again, both the convergence curve and the topology of the posterior distribution are presented.

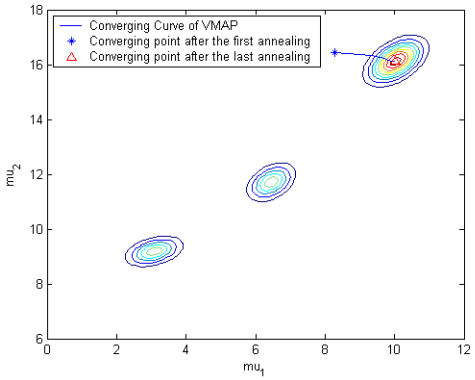
Some of the details of the experiments are described as follows. Firstly, the T_{max} is set to 200 and T_{min} is set to 0.01, where the annealing starts and ends respectively. Secondly, in



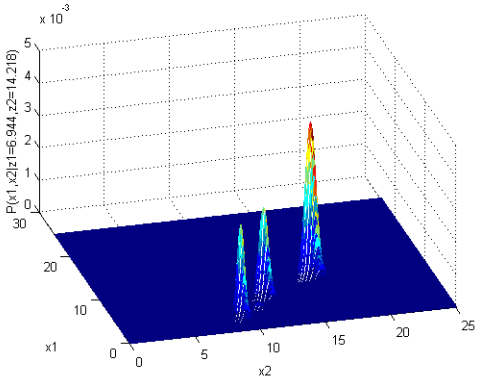
(a) $\mathbf{z}_1 = 10.0, \mathbf{z}_2 = 16.0$



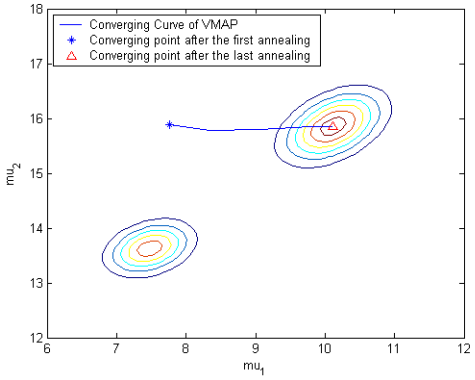
(b) $P(\mathbf{x}_1, \mathbf{x}_2|\mathbf{z}_1 = 10.0, \mathbf{z}_2 = 16.0)$



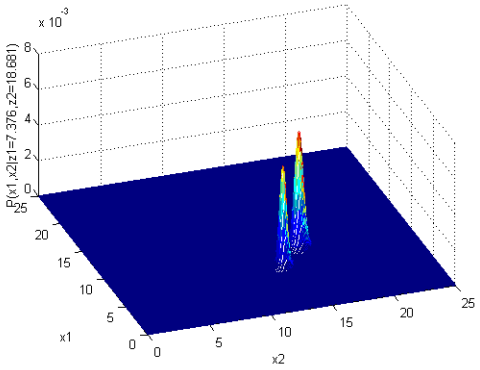
(c) $\mathbf{z}_1 = 6.944, \mathbf{z}_2 = 14.218$



(d) $P(\mathbf{x}_1, \mathbf{x}_2|\mathbf{z}_1 = 6.944, \mathbf{z}_2 = 14.218)$



(e) $\mathbf{z}_1 = 7.3762, \mathbf{z}_2 = 18.6813$



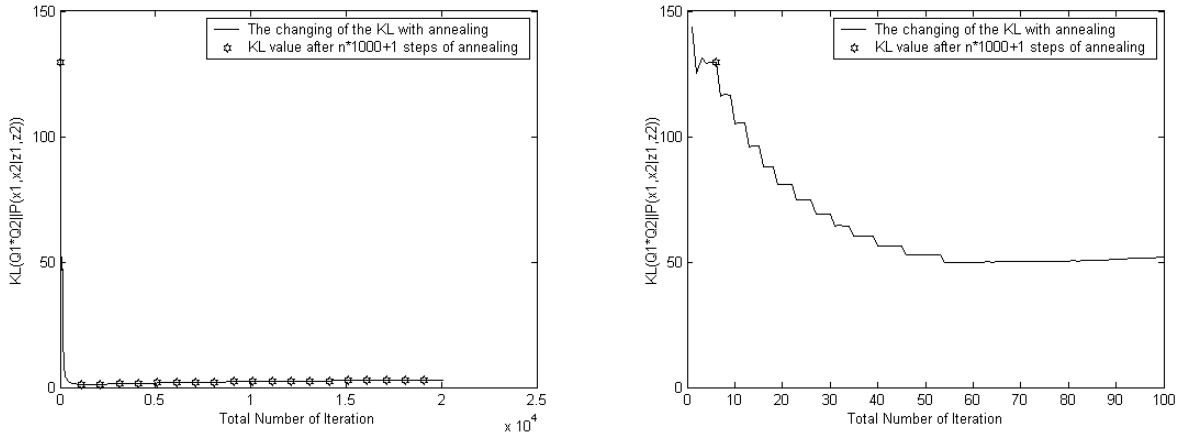
(f) $P(\mathbf{x}_1, \mathbf{x}_2|\mathbf{z}_1 = 7.3762, \mathbf{z}_2 = 18.6813)$

Figure 4.6. Convergence of the Variational MAP: (a) $* = \{9.60, 17.67\}$. $\triangle = \{12.68, 18.38\}$. The numerical global optimal $\{\mathbf{x}_1, \mathbf{x}_2\} = \{12.70, 18.40\}$; (c) $* = \{8.30, 16.45\}$. $\triangle = \{10.04, 16.13\}$. The numerical global optimal $\{\mathbf{x}_1, \mathbf{x}_2\} = \{10.00, 16.10\}$; (e) $* = \{7.76, 15.89\}$. $\triangle = \{10.11, 15.85\}$. The numerical global optimal $\{\mathbf{x}_1, \mathbf{x}_2\} = \{10.10, 15.80\}$.

each step of the annealing, we iterate Eq. 4.13 until convergence, i.e., we stop the updating of μ_1 and μ_2 if the difference between the updated value and the previous value is below the pre-specified threshold of 0.01.

Another concern would be about the convergence rate of the proposed variational MAP algorithm. Although a theoretical analysis of the convergence rate is very difficult, on the synthetic two-node problem, we generally observe that the first step of annealing takes the most number of iterations which ranges from 10 to 15 to converge, then in the following steps of annealing, it only takes 1 to 2 steps for the mean field iteration to converge. Therefore, empirically we achieve fast convergence of the proposed variational MAP algorithm. By the way, how to design the annealing scheme to achieve better result is also of interest just as we have mentioned in Sec. 4.5. However, a theoretic study of this problem seems to be a tremendous work. In all the experiments, we use the hyperbolical decreasing annealing scheme, i.e., $T = \frac{T_{max}}{K}$, it does achieve satisfactory results.

In fact, instead of manually setting a T_{min} for stopping the annealing, we can develop more rigorous criterion for the convergence of the annealing from the change of the $KL(\cdot)$. To make it clear, we plot the change of the $KL(\cdot)$ during the annealing of the experiment reported in Fig. 4.6(a) and Fig. 4.6(b), as shown in Fig. 4.7. From Fig. 4.7, we observe dramatic decrease of the $KL(\cdot)$ value in the approximately first 2000 round of annealing. Then the $KL(\cdot)$ will increase very slowly with the decreasing of T. The hexagons in the plot represents the $KL(\cdot)$ value after each 1000 round of annealing. Thus, there is one and only one global minimum $KL(\cdot)$ value during annealing in all the annealing steps. By checking the simulation results, we find that after the annealing which achieves the global minimum $KL(\cdot)$ value, the proposed variational MAP algorithm has already converged to the global maximum of the real posterior, e.g., in the experiments shown in Fig. 4.7, when the algorithm



(a) All the iterations

(b) First 100 iterations

Figure 4.7. The change of the $KL(Q_1(\mathbf{x}_1)Q_2(\mathbf{x}_2\|P(\mathbf{x}_1,\mathbf{x}_2|\mathbf{z}_1 = 10.0, \mathbf{z}_2 = 16.0)))$ during annealing. It is dramatically decreased in the first 1303 round of annealing and then increase very slowly in the following annealing process. The proposed variational MAP algorithm actually has converged to the global maximum of the real posterior distribution $P(\mathbf{x}_1, \mathbf{x}_2|\mathbf{z}_1 = 10.0, \mathbf{z}_2 = 16.0))$ at $\{\mu_1, \mu_2\} = \{12.6824, 18.3793\}$ after the 1303 round of annealing at $T = 0.1535$. The total number of the Gaussian constrained mean field iteration is 1420 up to the end of the 1303 annealing.

achieves the global minimum $KL(\cdot)$ value during the annealing, it has converged to the global MAP of the real posteriori distribution at $\{\mu_1, \mu_2\} = \{12.6824, 18.3793\}$, which corresponds to the 1303 round of annealing with $T = \frac{T_{max}}{1303} = \frac{200}{1303} = 0.1535$ and the total number of the mean field iteration is 1420. Actually, in the experiment, the running of the mean field iteration with annealing temperature after $T = 0.1535$ will not change μ_1 and μ_2 any more, it will just increase the $KL(\cdot)$ value a little bit since the Gaussian variational distribution tends to be more peaky.

Although we only show one plot of the change of the $KL(\cdot)$ value during annealing, all the experiments we have run showed the same pattern of the changes. Therefore, we conclude that we can stop the annealing when we find that after one step of annealing, the resulted $KL(\cdot)$ value is not less than the $KL(\cdot)$ value after the previous step of annealing. This also

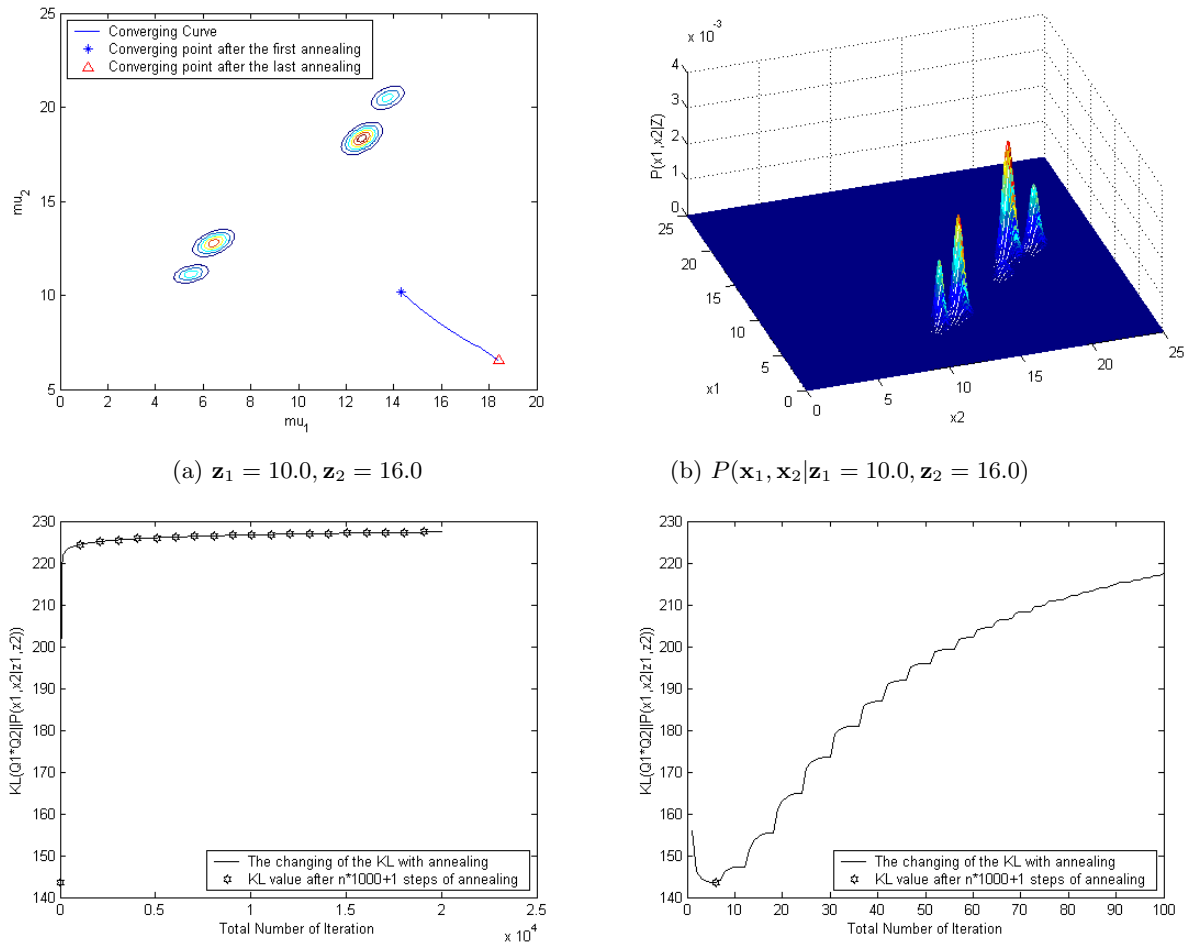
finds the optimal T_{min} which will result in the most efficient running of the algorithm. However, evaluating the $KL(\cdot)$ value may involve tremendous computation by itself. Therefore, we still tend to manually set the T_{max} and T_{min} to avoid the overhead introduced by the evaluation of the $KL(\cdot)$ value.

Under the same experimental setting, we also run the iteration of Eq. 4.15 on the same problem. Our observation is that the annealed iteration process does not converge at all. We show the experimental results when $\mathbf{z}_1 = 10.0, \mathbf{z}_2 = 16.0$ in Fig. 4.8. Fig. 4.8(a) shows the curve of the annealed iteration of Eq. 4.15, it failed to converge. Checking the value of the KL divergence during the iteration process, we find that it is increasing instead of decreasing with the iteration. We show the curve of the KL divergence in Fig. 4.8(c) while Fig. 4.8(d) presents the same curve in the first 100 iteration.

4.7.3. Variational MAP for tracking articulated human body

In this experiment, we implement the Monte Carlo simulation of the variational MAP algorithm for tracking articulated human body. We adopt the same Markov network to represent the articulated human body just as that in Chapter 3, where each body part is represented as a quad shape and the motion of each of them is represented as a probabilistic random variable in the 6 dimensional affine space. Please refer to Chapter 3 for the detailed description of the potential function $\psi_{ij}(\mathbf{x}_i, \mathbf{x}_j)$ and the observation function $\phi_i(\mathbf{z}_i|\mathbf{x}_i)$ of the Markov network.

Then the Monte Carlo version of the variational MAP algorithm is performed sequentially to recover the motion of the articulated human body from the video. Some of the sample result images are shown in Fig. 4.9. The proposed variational MAP algorithm recovers the



(a) $\mathbf{z}_1 = 10.0, \mathbf{z}_2 = 16.0$

(b) $P(\mathbf{x}_1, \mathbf{x}_2 | \mathbf{z}_1 = 10.0, \mathbf{z}_2 = 16.0)$

(c) All the iterations

(d) The first 100 iterations.

Figure 4.8. Annealed iteration of Eq. 4.15 in the 2-D illustrative synthesized problem: (a) $\mathbf{z}_1 = 10.0, \mathbf{z}_2 = 16.0$. $* = \{14.330, 10.148\}$. $\triangle = \{18.391, 6.5445\}$. The numerically global optimal is around $\{\mathbf{x}_1, \mathbf{x}_2\} = \{12.70, 18.40\}$; (b) $P(\mathbf{x}_1, \mathbf{x}_2 | \mathbf{z}_1 = 10.0, \mathbf{z}_2 = 16.0)$; (c) The KL value change in all the iterations. (d) The KL value change in the first 100 iterations.

articulated full-body motion very well across the video sequence, which has 767 frames. This is actually the annealed version of the MFMC algorithm proposed in [**46**, **126**].

For comparison, we also have implemented the MFMC algorithm [**126**] and the multiple independent tracker which has been used as a comparison of the MFMC algorithm in [**126**]. Our experimental results reveal that the MFMC algorithm can track the articulated motion

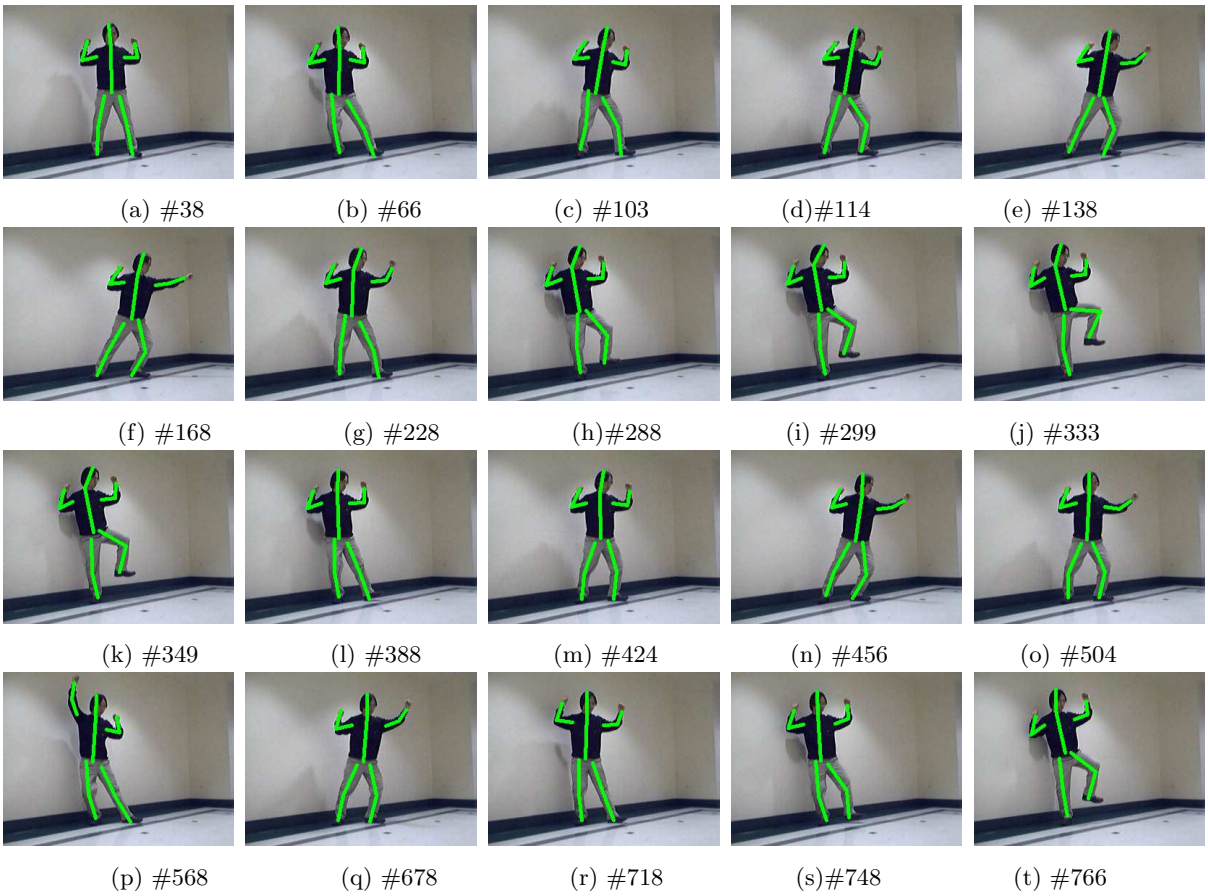
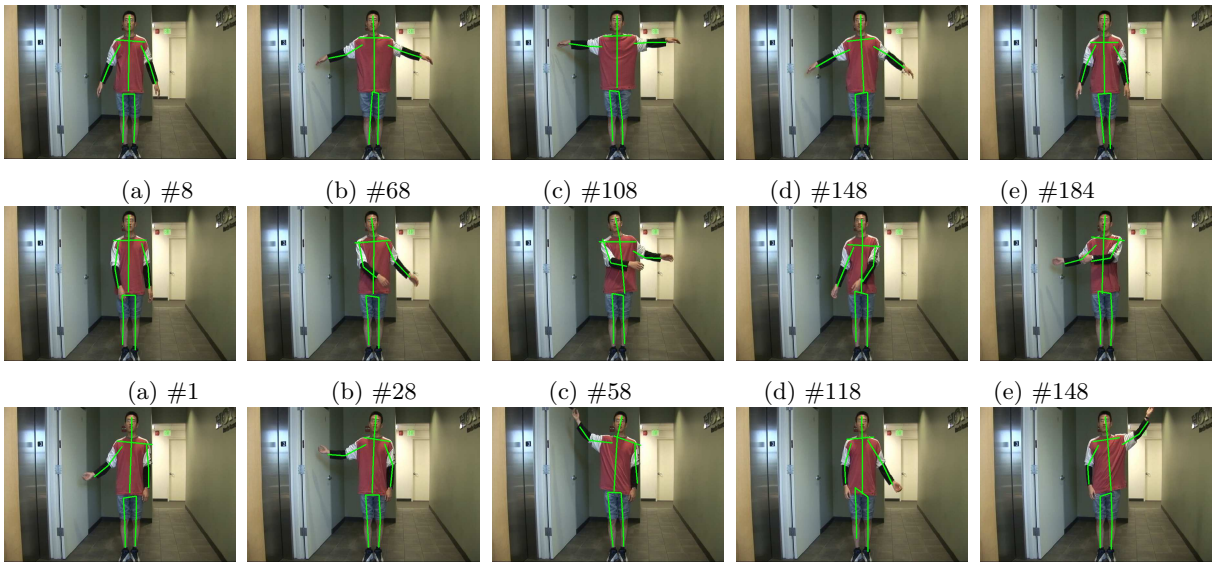


Figure 4.9. Variational MAP for tracking articulated human body, the video sequence has 767 frames and it can robustly recover the full human body motion across the whole sequence.

well until the 368th frame and it loses track after that. The reason for the tracking failure of the MFMC algorithm is that the heavy multi-modality of the motion posterior causes the mean estimate to be significantly deviated from the MAP estimate of the motion. Thus it could hardly indicate the true motion, e.g., as we can observe in frame #370. It will make the online estimated dynamical model to be not accurate, which in turn causes tracking failure. Also, just as reported in [**126**], the multiple independent tracker loses track from the start.



(a) #8 (b) #68 (c) #108 (d) #148 (e) #184

(a) #1 (b) #28 (c) #58 (d) #118 (e) #148

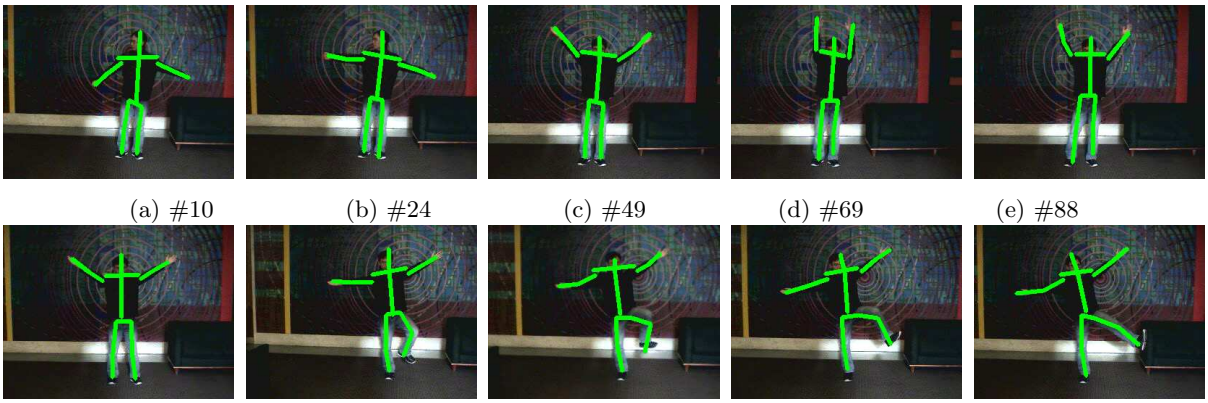
(a) #8 (b) #38 (c) #98 (d) #208 (e) #268

Figure 4.10. Tracking 10-part full body by variational MAP: **First row**-Clap sequence; **Second row**-Swing sequence; **Third row**-Toss sequence. Frame numbers are indicated in the bottom of the result image.

When comparing the variational MAP algorithm with the MFMC algorithm, we set all the parameters of the potential functions $\psi_{ij}(\mathbf{x}_i, \mathbf{x}_j)$ and the observation functions $\phi_i(\mathbf{z}_i|\mathbf{x}_i)$ to be the same for both algorithms. Because of the annealing process, the proposed variational MAP algorithm needs more mean field iterations than the MFMC algorithm. Our experiments show that only the first step of annealing needs more iterations, in the following steps of annealing, it generally needs less than half of the mean field iterations of the MFMC algorithm. So the variational MAP algorithm only increases the computation linearly in comparison with the MFMC algorithm. Therefore, based on the analysis of the complexity of the MFMC algorithm [**46**, **126**], the variational MAP algorithm also achieves linear complexity with respect to the number of body parts in tracking the articulated human body.

Here we present some more tracking results on three video sequences, in which a person performs three actions such as "clap", "swing" and "toss". The video sequences have 186, 216 and 335 frames respectively. These video sequences are more challenging due to the self occlusion between the limbs and torso. The variational MAP algorithm obtains robust results and sample result images are presented in Fig. 4.10. For clarity, we only overlay the skeleton of the quadrangle shapes of the recovered articulated motion on the sample images.

The last video sequence we have tested is a full-body video sequence of 212 frames, where the background is more cluttered. We present sample result images in Fig. 4.11. As we can observe, the background wall behind the moving person is quite cluttered. Although the variational MAP algorithm successfully recovers the motion from the video, the results are less smooth than the results obtained when the background is clean. The reason is that in our current implementation, the motion prior encoded in the hidden layer of the Markov network is a zero mean Gaussian to reinforce the spatial coherence constraints. As we have discussed, this prior is a weak one. Therefore, the image likelihood functions must be reasonable good for the proposed approach to obtain good results. Although we build the image likelihood functions from both edge and intensity cues to make it more robust, the cluttered background may still degrade the quality of the adopted image likelihood function, and thus degrade the quality of the tracking results. Solution to this degradation issue due to clutter may be building better image likelihood functions. For example, when the background is static, we can perform background subtraction firstly to remove the background clutters. We must emphasize here that none of the experimental results reported here used any background subtraction techniques. When the background is not static, we may build more discriminative image likelihood functions using more image cues such as textures and color distributions. We defer that to our future work.



(a) #10 (b) #24 (c) #49 (d) #69 (e) #88

(f) #106 (g) #145 (h) #168 (i) #184 (j) #203

Figure 4.11. Tracking 10-part full body. The background present significant clutter. Frame numbers are indicated in the bottom of the result image.

All the algorithms are implemented using C++, no code optimization is performed. They are running in a 2.5 GHz PC under Windows XP. We design 6 annealing steps and in the first step of the annealing, we iterate the mean field fixed-point equations for 6 times and in the following annealing steps, we run the mean field fixed-point equations for 3 times. The algorithm can thus run at the speed of 0.2 frames per second. While in the MFMC algorithm, we iterate the mean field fixed-point equation 6 times and the mean values of the recovered mean field distribution are adopted as the tracking result. It can roughly run at the speed of 0.6 frames per second, just similar to what has been reported in Chapter 3. We also use 200 particles for each of the body parts.

Another issue of interest would be that if using one control parameter T for all the different component of the state random variable $\mathbf{x}_i$ is a good setting. In theory, it will have no problem, but in real experiments, it may encounter problems since different components of $\mathbf{x}_i$ may have different ranges. For example, in the 6 dimensional affine motion space, the translation component and the scaling component have different ranges. Thus we design

different annealing schemes for different component of the affine state vector, i.e., the annealing of the translation components of $\mathbf{x}_i$ starts at $\mathbf{T}_{max1} = 8$ and the annealing of the scaling components of $\mathbf{x}_i$ starts at $\mathbf{T}_{max2} = 0.6$.

4.8. Conclusion and future work

This chapter proposes a novel variational MAP algorithm for the optimal MAP estimation of complex stochastic systems. By constraining the mean field variational distribution to be multivariate Gaussian, a DA scheme is naturally incorporated to the mean field variational analysis to pursue the optimal MAP estimation. Our main contributions are:

(1) We show that the limit of the topology of the KL divergence between a multivariate Gaussian distribution $g(\mathbf{X}) = \mathcal{N}(\mathbf{X}|\tilde{\mu}, \sigma^2\mathcal{I})$ and an arbitrary p.d.f. $p(\mathbf{X})$, when the σ^2 approaches to zero, will converge to the topology of $-log(\tilde{\mu})$(see Lemma 2 in Appendix A). Thus there is an one-to-one correspondence of the minima between the $\lim_{\sigma^2 \to \mathbf{0}} KL(g(\mathbf{X})||p(\mathbf{X}))$ and the maxima of the p.d.f. $p(\mathbf{X})$, and the limit of the infimum point of the KL divergence will converge to the supreme point of the $p(\mathbf{X})$, as shown in Theorem 4.3.1.

(2) Based on Theorem 4.3.1, we nicely incorporate a DA scheme into the Gaussian constrained mean field variational analysis to pursue the optimal MAP estimation of complex stochastic systems. Although DA may not guarantee global optimality, our extensive synthetic and real experiments show that it is very likely to achieve a global or near global optimal result. Therefore, we achieve an efficient and effective way for optimal MAP estimation.

There are also several questions need to be further investigated:

(1) Although we have empirically shown that the mean field fixed-point iteration in Eq. 4.11 and 4.12 is superior to the iteration in Eq. 4.14 and 4.15, i.e., the latter two failed to converge even in a relative simple synthetic problem, we are still interested in investigating theoretically why the former two equations can obtain better results under the context of optimization.

(2) Is there an optimal annealing scheme which can guarantee to achieve the optimal results more efficiently? The answer of this question will also reveal the convergence rate of the annealing scheme.

(3) When will the proposed variational MAP algorithm achieve the global optimality? Although generally in our experiments, the variational MAP algorithm with the hyperbolic decreasing DA scheme achieves good results, practically there is no guarantee that the algorithm will achieve global optimality. Should there be a sufficient condition, or a necessary condition, or both for the global optimality?

(4) With regarding to applying the proposed variational MAP algorithm to computer vision problems, should there be an efficient way of incorporating some bottom-up processing to facilitate more efficient convergence of the algorithm? The answer of this question will achieve a data driven variational MAP algorithm for many computer vision problems.

We will further study the above questions in our future work.

CHAPTER 5

Data driven belief propagation for human pose estimation

To automatically bootstrap the collaborative motion analyzer, e.g., to automatically initialize the articulated body tracking algorithm, we need to recover the motion parameters from single images. That is, we no longer have the dynamic priors to be integrated into the proposed collaborative approach. This chapter present how we can incorporate a data driven scheme into the proposed collaborative approach to effectively achieve this goal, which nicely combines *bottom-up* reasoning and *top-down* reasoning in a unified way. Again, to be concrete, we use articulated body as an example.

5.1. Introduction

Inferring human pose from single images is arguably one of the most difficult problems in computer vision and finds numerous applications from motion analysis to action recognition. In this chapter, we posit the 2-D human pose estimation problem within a probabilistic framework and develop an inference algorithm on a rigorous statistical footing. A human body pose is modeled by a Markov network where the nodes denote body parts and the edges encode constraints among them. An efficient data driven belief propagation Monte Carlo algorithm with importance sampling functions, built from low-level visual cues, is proposed to infer the 2-D human pose from a single image snapshot.

From a set of manually labeled images, we apply principal component analysis to learn the 2-D shape models of each body part which serve as prior knowledge in predicting potential

candidates. Each body part is represented by a state variable describing its shape and location parameters. The data driven importance sampling for the head pose is built using a computationally efficient AdaBoost-based face detector [**118**]. Constrained by the head location from face detection, a probabilistic hough transform [**66**] is adopted to extract salient line segments in the image and they are assembled to form good candidates for constructing an importance sampling function for the human torso. A skin color model pertaining to the specific subject in the image is built based on the face detection result, and is utilized in sampling functions to predict potential body part candidates such as arms and legs.

The data driven importance functions for body parts are incorporated in the belief propagation Monte Carlo framework for efficient Bayesian inference of the human pose. For human pose estimation, the observation models are built based on the steered edge response of the predicted body parts. Diametric to the sequential data driven Markov chain Monte Carlo algorithm, the proposed algorithm integrates both top-down as well as bottom-up reasoning mechanism with visual cues, and carries out the inference tasks in parallel within a sound statistical framework. For concreteness, we apply the developed method to estimate pose of soccer players in single images with cluttered backgrounds. Experimental results demonstrate the potency and effectiveness of the proposed method in estimating human pose from single images.

The organization of the rest of the chapter is as follows: in Sec. 5.2, related work are summarized and discussed. We then present our articulated human body model in Sec. 5.3. The data driven belief propagation algorithm (DDBP), along with means of building importance functions from bottom-up, are presented in Sec. 5.4. Validation experimental results are shown and discussed in Sec. 5.5. We conclude with discussions on limitations of the current work and future plan to tackle these problems in Sec. 5.6.

5.2. Prior work and context

While there exist numerous works on human body tracking [**81**], only a few of them address the initialization problem, i.e., estimating the human pose from single or multiple views. We observe the emergence of research work on this topic with impressive results [**25**, **69**, **80**] in the last few years. These algorithms are categorized into deterministic and statistical methods for ease of presentation.

Deterministic methods either approach this problem by applying deterministic optimization methods where the objective function is the matching error between the model and the image data [**5**, **25**] or between the image data and the exemplar set [**101**]. An alternative is to build detectors for different body parts and rank the assembled configuration based on a set of human coded criteria [**80**]. Notwithstanding the demonstrated success, there exist many challenging issues to be resolved for robust and efficient pose estimation. First, it entails solving an optimization problem of high dimensionality and thus the computation is inevitably intractable unless certain assumptions are explicitly made. Consequently, the application domains are limited to uncluttered backgrounds [**5**, **25**] or the human body of fixed scale [**80**]. Second, the set of exemplars must be large enough to cover the parameter space to achieve satisfactory estimation results at the expense of growing computational complexity [**101**]. Third, it is difficult to build robust body part detectors except faces [**118**] due to the large appearance variation cased by clothing [**80**].

One salient merit of statistical formulation for posture estimation is that prior knowledge of human body parts (e.g., appearance, shape, edge and color) can all be exploited and integrated in a rigorous probabilistic framework for efficient inference. Ioffe and Forsyth [**49**]

propose an algorithm to sequentially draw samples of body parts and make the best prediction by matching the assembled configurations with image observations. However, it is best applied to estimate poses of humans in images without clothing or cluttered background since their method relies solely on edge cues. Sigal et al. [**106**] resort to a non-parametric belief propagation algorithm [**51**, **110**] for inferring the 3-D human pose as the first step of their human tracking algorithm. Background subtraction and images from multiple views are employed to facilitate the human pose estimation and tracking problems. Lee and Cohen [**69**] apply the data driven Markov Chain Monte Carlo (DDMCMC) algorithm [**116**] to estimate 3-D human pose from single images, in which the MCMC algorithm is utilized to traverse the pose parameter space. Nevertheless it is not clear how the detailed balance condition and convergence in the Markov chain are ensured. Most importantly, the problem of inferring 3-D body pose from single 2-D images is intrinsically ill-posed as a result of depth ambiguity.

In this work [**133**], we propose a statistical formulation to infer 2-D body pose from single images. Different from the previous works, the proposed algorithm integrates the top-down and bottom-up inference with visual cues through a data driven belief propagation Monte Carlo algorithm for Bayesian reasoning. The algorithm is intrinsically parallel which is in direct contrast to the sequential sampling algorithm [**49**] and the sequential DDMCMC approach [**69**]. Furthermore we explicitly learn the shape models of body parts using quadrangles rather than rectangular templates [**25**, **49**, **80**], thereby facilitating inference of pose parameters.

5.3. Bayesian formulation

We posit the human pose estimation problem within a Bayesian framework and the task is to recover the hidden states, i.e., pose parameters, from image observations.

5.3.1. Markov network

A human body configuration is represented by a Markov network as shown in Figure 5.1. Each random variable $\mathbf{x}_i$ represents the pose parameter (i.e., hidden state) of body part i, e.g., $\mathbf{x}_h$ describes the pose of *head*, $\mathbf{x}_t$ describes the pose of *torso*, and $\mathbf{x}_{rul}$ describes the pose of the *right-upper-leg*. Each undirected link models the constraints between two adjacent body parts by a potential function $\psi_{ij}(\mathbf{x}_i, \mathbf{x}_j)$. Each directed link depicts the image observation $\mathbf{z}_i$ of body part i with an observation likelihood function $\phi_i(\mathbf{z}_i|\mathbf{x}_i)$. Let $\mathcal{S}$ be the set of all subscripts, we denote the set of pose parameters $\mathbf{X} = \{\mathbf{x}_i, i \in \mathcal{S}\}$ and the set of observations $\mathbf{Z} = \{\mathbf{z}_i, i \in \mathcal{S}\}$, respectively. The joint posterior distribution of this Markov network is

$$P(\mathbf{X}|\mathbf{Z}) \propto \prod_{(i,j)\in\mathcal{E}} \psi_{ij}(\mathbf{x}_i, \mathbf{x}_j) \prod_{i\in\mathcal{V}} \phi_i(\mathbf{z}_i|\mathbf{x}_i), \tag{5.1}$$

where $\mathcal{E}$ is the set of all undirected links and $\mathcal{V}$ is the set of all directed links [**58**]. Consequently, the pose estimation problem is formulated as a Bayesian inference problem of estimating the marginal posterior distribution $P(\mathbf{x}_i|\mathbf{Z})$.

A brute force approach to computing Eq. 5.1 is intractable since it involves numerous integrals of real valued random variables in every $P(\mathbf{x}_i|\mathbf{Z})$. The belief propagation algorithms, facilitated by local message passing (i.e., local computation), offer an efficient solution to such inference problems. Recently a Monte Carlo approach for belief propagation is proposed to deal with graphical models with non-Gaussian distributions [**41**]. In Sec. 5.4, we present

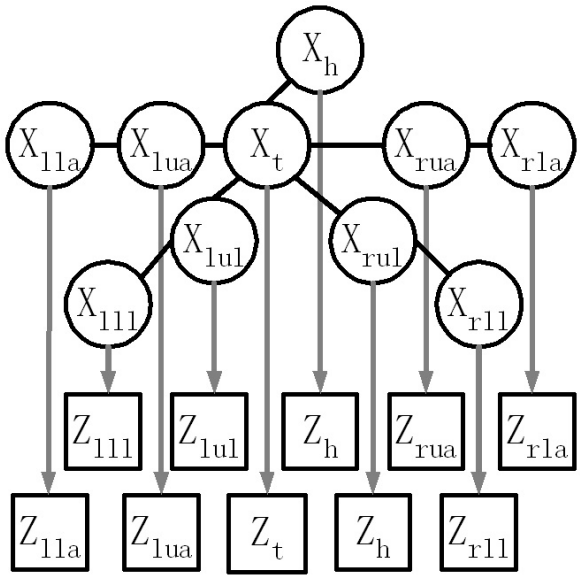


Figure 5.1. Markov network for human body pose.

a novel data driven belief propagation algorithm, which naturally integrate the bottom-up reasoning with the belief propagation Monte Carlo algorithm in a principled way.

5.3.2. Pose parametrization

We represent each body part by a quadrangular shape in a way similar to the existing works [**25**, **80**]. However, we do not model them with rectangles or trapezoids since the body contours usually do not form parallel lines in images. From a set of 50 images, we manually labeled the quadrangular shapes and poses of human parts which best match human perception. A few examples of the labeled images are illustrated in Fig. 5.2.

For each of the labeled quadrangular shape, we define the lines along the body outer contour as the *left* and the *right* lines, and the other two lines as the *top* and the *bottom* lines, respectively. We define the local coordinate system of each body part by choosing the centroid of the quadrangular shape as its origin. The **Y** axis is pointed from the middle

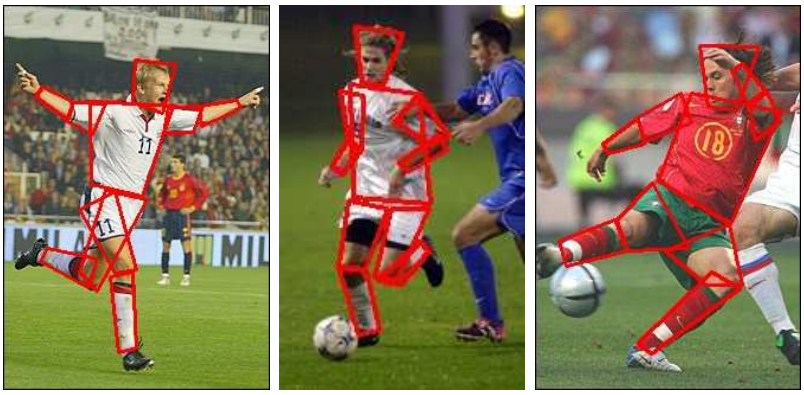


Figure 5.2. Examples of labeled images.

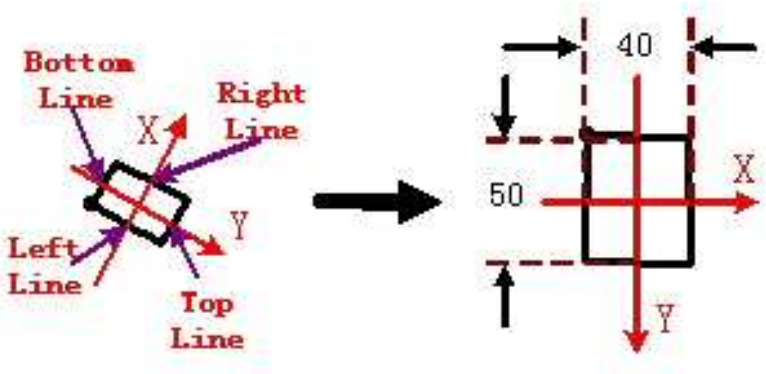


Figure 5.3. Normalization of the labeled shape.

point of the *top* line to the middle point of the *bottom* line, and the **X** axis is perpendicular to the **Y** axis such that the local coordinate system is only subject to a rotation and a translation of the image coordinate system. Each labeled shape is then rotated with respect to a reference frame and then normalized in both **X** and **Y** directions, i.e., the length (width) along the **X** axis between the *left* and the *right* lines is normalized to 40 pixels, and the length (height) along the **Y** axis between the *top* and the *bottom* lines is normalized to 50 pixels, as depicted in Fig. 5.3. Each normalized shape is then represented by a 8-dimensional vector by clockwise enumerating the coordinates of the four vertices.

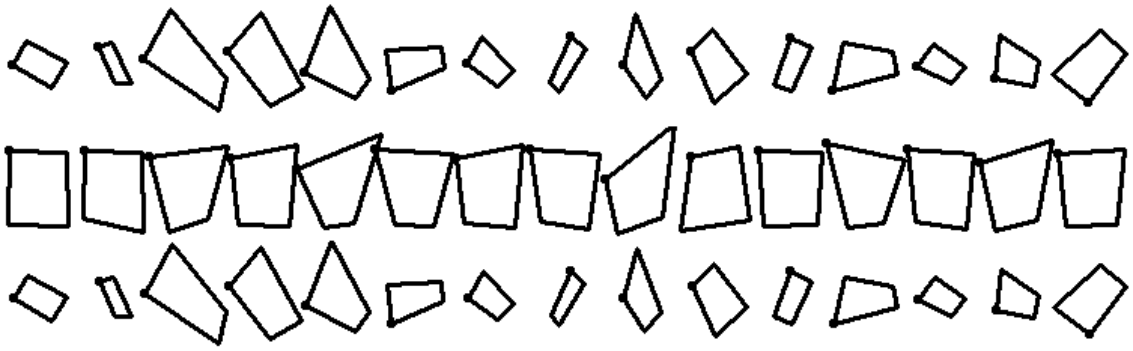


Figure 5.4. The original shapes (first row), the normalized (second row), and the reconstructed (third row) shapes of the *right-upper-arm* using probabilistic PCA.

We apply probabilistic principal component analysis (PCA) [**115**] to each set of the 8-dimensional normalized body part shapes for dimensionality reduction. In Sec. 5.4.2, we show how we use the learned shape model with probabilistic PCA to construct good importance sampling functions for body parts. In our experiments, 99% of the shape variation can be retained with the top 3 principal components. We denote the shape representation with reduced dimensionality for each body part $i \in \mathcal{S}$ as $\mathbf{ps}_i$. Consequently, the 2-D pose of body part i can be represented by the rotation θ, scaling $\mathbf{s}_x$, $\mathbf{s}_y$, and translation $\mathbf{t}_x$, $\mathbf{t}_y$, in both $\mathbf{X}$ and $\mathbf{Y}$ directions of $\mathbf{ps}_i$, i.e.,

$$\mathbf{x}_i = \{\mathbf{ps}_i, \mathbf{s}_x, \mathbf{s}_y, \theta, \mathbf{t}_x, \mathbf{t}_y\}. \tag{5.2}$$

where we call $\mathbf{ps}_i$ the intrinsic pose parameter and the rest the extrinsic pose parameters. By learning a low-dimensional shape representation, we reduce the originally 13-dimensional state space to 8 dimensions which in turns facilitates efficient sampling process. Fig. 5.4 shows some of the original labeled shapes, the normalized shapes, as well as the reconstructed shapes from the probabilistic PCA for the *right-upper-arm*. It is clear that the reconstructed shapes match well with the original labeled shapes.

5.3.3. Potential function and likelihood model

As mentioned earlier, a potential function ψ_{ij} models the pose constraints between two adjacent body parts. For pose estimation, the natural constraints entail any two adjacent body parts should be *loosely connected* [**106**], and we use a Gaussian distribution to model the Euclidean distance between the link points of two adjacent body parts, i.e.,

$$\psi_{ij}(\mathbf{x}_i, \mathbf{x}_j) \propto \exp\left(-\frac{\|\tilde{\mathbf{P}}\mathbf{t}_{ij} - \tilde{\mathbf{P}}\mathbf{t}_{ji}\|^2}{2\sigma_{ij}^2}\right). \tag{5.3}$$

where $\|\cdot\|$ is the Euclidean distance, σ_{ij} is the variance learned from the manually labeled images, and $\tilde{\mathbf{P}}\mathbf{t}_{ij}$ is the link point of the i^{th} to j^{th} body part while $\tilde{\mathbf{P}}\mathbf{t}_{ji}$ is the link point of the j^{th} to i^{th} body part. Fig. 5.5 shows all the link points of the body parts. In our model, the link points are either corner points or middle points of either *bottom* or *top* line of the shape. For example, the link point of the *left-upper-arm* to the torso is defined as the corner point of the *left* line and the *bottom* line of the left-upper-arm shape, and the link point of the torso to the left-upper-arm is also specified by the corner point of the *left-bottom* corner of the torso shape. Whereas the link point of the *left-upper-arm* to the *left-lower-arm* is delineated by the middle point of the *top* line of the *left-upper-arm* shape, the link point of the *left-lower-arm* to the *left-upper-arm* is defined as the middle point of the *bottom* line of the *left-lower-arm* shape.

Although object appearance or texture has been successfully utilized in tasks such as face detection, the body contour information may be the only salient cue at our disposal as clothing causes large visual variation. In this work, the likelihood function ϕ_i is constructed based on the average steered edge response [**103**] along the boundaries of the pose hypothesis of an body part. For example, let the rotation angle of one line segment l be α and the total

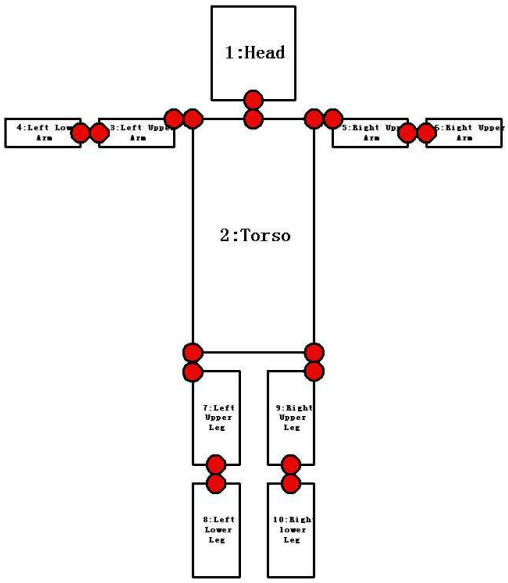


Figure 5.5. Each pair of red circle points represents the link point pair of two adjacent body parts. The link points are either corner points or middle points of bottom or top lines.

number of points on the line is N_l, then the average steered edge response is

$$\bar{\mathcal{E}}_{l,\alpha} = \frac{1}{N_l \mathcal{E}_m} \sum_{(\mathbf{x}_i, \mathbf{y}_i) \in l} |\mathcal{E}_x(\mathbf{x}_i, \mathbf{y}_i) \sin\alpha - \mathcal{E}_y(\mathbf{x}_i, \mathbf{y}_i) cos\alpha|, \tag{5.4}$$

where $\mathcal{E}_m$ is the maximum value of the steered edge response. Unlike [**103**], we do not compute the steered edge response at different scales because the average steered edge responses across scales may make the steered edge response less discriminant.

Instead, we compute the steered edge response in the RGB channels, i.e., $\mathcal{E}_\alpha^{(R)}(\mathbf{x}_i)$, $\mathcal{E}_\alpha^{(G)}(\mathbf{x}_i)$ and $\mathcal{E}_\alpha^{(B)}(\mathbf{x}_i)$ for each hypothesized body part $\mathbf{x}_i$. For *head* and *torso*, the average steered edge response is computed using all the four line segments of the shape pose hypothesis, whereas the average steered edge response is only calculated on the *left* and *right* line segments for the other body parts. Since all the steered edge responses have been normalized between 0 and 1, the likelihood function is defined based on the maximum steered

edge response, i.e.,

$$\phi_i(\mathbf{z}_i|\mathbf{x}_i) = \max(\mathcal{E}_\alpha^{(R)}(\mathbf{x}_i), \mathcal{E}_\alpha^{(G)}(\mathbf{x}_i), \mathcal{E}_\alpha^{(B)}(\mathbf{x}_i)). \tag{5.5}$$

The reason for using the maximum steered edge response from different color channels is based on our empirical studies in which more discriminant likelihood functions can be obtained using the maximum rather than average edge response. We have experimented with the Gibbs likelihood model proposed in [**100**] but the performance is less satisfactory. One explanation is that background subtraction is utilized so that the body contours can be better extracted before learning a Gibbs model for likelihood estimation [**100**]. Nevertheless, background subtraction is inapplicable in this work as we aim to estimate human pose from single images.

5.4. Data driven belief propagation

With the Bayesian formulation described in Sec. 5.3, the pose estimation problem is to infer the marginal posterior distribution. In this section, we propose a data driven belief propagation Monte Carlo algorithm (DDBPMC) for Bayesian inference on real valued graphical models.

5.4.1. Belief propagation Monte Carlo

Belief propagation is an efficient algorithm to compute posterior, $P(\mathbf{x}_i|\mathbf{Z})$, through a local message passing process where the message from $\mathbf{x}_j$ to $\mathbf{x}_i$ is computed by [**29**, **58**]:

$$\mathbf{m}_{ij}(\mathbf{x}_i) \leftarrow \int_{\mathbf{x}_j} \phi_j(\mathbf{z}_j|\mathbf{x}_j)\psi_{ij}(\mathbf{x}_i, \mathbf{x}_j) \prod_{k \in \mathcal{N}(j)\setminus i} \mathbf{m}_{jk}(\mathbf{x}_j), \tag{5.6}$$

where $\mathcal{N}(j)\backslash i$ is the set of neighboring nodes of $\mathbf{x}_j$ except $\mathbf{x}_i$. The belief propagation algorithm iteratively updates the messages passed among the connected nodes until it converges, and the marginal posterior distribution $P(\mathbf{x}_i|\mathbf{Z})$ on node $\mathbf{x}_i$ can be efficiently computed by

$$P(\mathbf{x}_i|\mathbf{Z}) \propto \phi_i(\mathbf{z}_i|\mathbf{x}_i) \prod_{j\in\mathcal{N}(i)} \mathbf{m}_{ij}(\mathbf{x}_i). \tag{5.7}$$

When both the potential function $\psi_{ij}(\mathbf{x}_i, \mathbf{x}_j)$ and the observation likelihood $\phi_i(\mathbf{z}_i|\mathbf{x}_i)$ are Gaussian distributions, Eq. 5.6 can be evaluated analytically and thus Eq. 5.7 can be analytically computed. However, situations arise where the observation likelihood functions $\phi_i(\mathbf{z}_i|\mathbf{x}_i)$ can only be modeled with non-Gaussian distributions. In such cases, the messages $\mathbf{m}_{ij}(\mathbf{x}_i)$ are also non-Gaussians, thereby making the computation intractable.

To cope with this problem and allow greater flexibility, we resort to Monte Carlo approximation within the belief propagation formulation, and thereby a belief propagation Monte Carlo (BPMC) algorithm. We represent both the message $\mathbf{m}_{ij}(\mathbf{x}_i)$ and the marginal posterior distribution $P(\mathbf{x}_i|\mathbf{Z})$ as weighted sample sets by

$$\mathbf{m}_{ij}(\mathbf{x}_i) \sim \{\mathbf{s}_i^{(n)}, \omega_i^{(j,n)}\}_{n=1}^N, j \in \mathcal{N}(\mathbf{x_i}) \tag{5.8}$$

$$P(\mathbf{x}_i|\mathbf{Z}) \sim \{\mathbf{s}_i^{(n)}, \pi_i^{(n)}\}_{n=1}^N. \tag{5.9}$$

The iterative computation in the belief propagation can be implemented based on these weighted sample sets as summarized in Fig. 5.6.

Note that in both the non-parametric belief propagation [**110**] and PAMPAS [**51**] algorithms, the messages as well as the marginal distributions are modeled with Gaussian mixtures, and the message passing process is carried out by a Markov chain Monte Carlo

Monte Carlo simulation of the belief propagation algorithm

(1) `Importance Sampling:` Sample $\{\mathbf{s}_{i,k+1}^{(n)}\}_{n=1}^{N}$ from an importance function $I_i(\mathbf{x}_i)$.

(2) `Re-weight:` For each $\mathbf{s}_{i,k+1}^{(n)}$ and each $j \in \mathcal{N}(i)$, set $\omega_{i,k+1}^{(j,n)} = G_i^{(j)}(\mathbf{s}_{i,k+1}^{(n)})/I_i(\mathbf{s}_{i,k+1}^{(n)})$ where

$$G_i^{(j)}(\mathbf{s}_{i,k+1}^{(n)}) = \sum_{m=1}^{N} \left[\pi_{j,k}^{(m)} \phi_j(\mathbf{z}_{j,k}^{(m)}|\mathbf{s}_{j,k}^{(m)}) \psi_{ji}(\mathbf{s}_{j,k}^{(m)}, \mathbf{s}_{i,k+1}^{(n)}) \prod_{l \in \mathcal{N}(j)\setminus i} \omega_{j,k}^{(l,m)} \right] \tag{5.10}$$

(3) `Normalization:` Normalize $\omega_{i,k+1}^{(j,n)}, j \in \mathcal{N}(i)$, then set and normalize

$$\pi_{i,k+1}^{(n)} = p_i(\mathbf{z}_{i,k+1}^{(n)}|\mathbf{s}_{i,k+1}^{(n)}) \prod_{l \in \mathcal{N}(i)} \omega_{i,k+1}^{(l,n)}$$

(4) `Iteration:` $k \leftarrow k+1$, iterate $1 \rightarrow 4$ until convergence.

(5) `Inference:` $p(\mathbf{x}_i|\mathbf{Z}) \sim \{\mathbf{s}_i^{(n)}, \pi_i^{(n)}\}_{n=1}^{N}$ where $\mathbf{s}_i^{(n)} = \mathbf{s}_{i,k}^{(n)}$ and $\pi_i^{(n)} = \pi_{i,k}^{(n)}$.

Figure 5.6. Data driven belief propagation.

(MCMC) algorithm. In contrast, the BPMC algorithm models both the messages and marginal distributions with weighted samples, and the message passing process is computed efficiently based on the samples drawn from an importance sampling. It is worth emphasizing that good importance functions leads to efficient computation in the BPMC algorithm and better inference results. In Sec. 5.4.2, we show how we construct good importance functions with bottom-up visual cues for human pose estimation.

5.4.2. Data driven importance sampling

In this section, we describe the importance functions for drawing samples of body parts using different visual cues. For concreteness, we present our algorithm with an application to estimate pose of soccer players in images. In such cases, we can exploit certain image cues for computational efficiency.

(a) (b) (c) (d)

Figure 5.7. (a). Face detected by a AdaBoost-based face detector. (b). Image specific skin color segmentation. (c). Fitted lower-arm and upper-leg hypotheses. (d) Upper-arm and lower-leg hypotheses (yellow quadrangular shape).

5.4.2.1. Importance function for head pose. With the demonstrated success in detecting faces efficiently, we utilize a variant of the AdaBoost-based face detector [**118**] to locate the face of a human in an image. However, this view-based detector performs best in detecting faces in upright frontal views although this problem can be alleviated by utilizing a multi-view extension. Figure 5.7(a) shows one face detected by the AdaBoost-based detector.

One common problem with this view-based face detector is that the raw detection results are usually not very accurate (i.e., the returned rectangles do not precisely lock on faces in the correct pose and often enclose background pixels), and thus more efforts are required to better estimate head pose. Since skin color pixels account for the majority portion of a rectangular area enclosing a face, we use a k-means algorithm ($k = 2$) to group the pixels within the rectangle into skin/non-skin clusters. The center of the face rectangle is repositioned to centroid of the cluster of skin color pixels. We then project the rectangular shape onto the learned PCA subspace of the head shape, thereby obtaining its intrinsic pose parameters as defined in Eq. 5.2. Along with the extrinsic rotation, scaling and translation

parameters extracted from the face rectangle, we obtain an approximated head pose $\mathbf{Ix}_h$, and thereby an importance sampling function:

$$\mathbf{I}_h(\mathbf{x}_h) \sim \mathcal{N}(\mathbf{x}_h|\mathbf{Ix}_h, \mathbf{\Sigma}_h) \tag{5.11}$$

where $\mathbf{\Sigma}_h$ is a diagonal covariance matrix.

5.4.2.2. Importance functions for arm and leg pose. For the human pose estimation problem considered in this chapter, the soccer players often wear short sleeve shirts and short trunks, and consequently skin color is a salient cue for locating *lower-arm* and *upper-leg* regions.

A skin color model is constructed from the pixels of skin color cluster obtained from the k-means algorithm within the detected face region as discussed in Sec. 5.4.2.1. Specifically, a 2-D color histogram is computed from the normalized RGB pixel values of the skin color cluster. Although it is difficult and time consuming to develop a generic skin color model to account for all variations (as a result of lighting and race factors), it is relatively easy and effective to construct a skin color model specific to the human subject considered for pose estimation, and consequently skin color regions can be extracted effectively with thresholds. Fig. 5.7(b) shows some segmentation results using the learned skin color histogram, and Fig. 5.7(c) shows the results with best fit rectangles after discarding small blobs. Note that the number of skin tone blobs do not necessarily match the number of body parts. Geometric cues such as shape, size, position, and orientation with respect to the head position of a human can be exploited to generate good pose hypotheses for the *lower-arm* and the *upper-leg* body parts from these fitted rectangles. The hypotheses for the *upper-arm* and the *lower-leg* are then generated by first rotating the shape with respect to the link point of the

corresponding *lower-arm* and the *upper-leg* hypotheses respectively, and then evaluating the image likelihoods based on edge response using Eq. 5.4 and Eq. 5.5 for each rotation angle. The hypotheses with maximum likelihoods for *upper-arm* and *lower-leg* parts are selected for importance functions. Fig. 5.7(d) shows one hypothesis for each of the upper-arm and lower-leg. The importance sampling function is modeled by a Gaussian mixture of these hypotheses. That is, after obtaining $\mathcal{K}$ good pose hypothesis $\mathbf{Ix}_i^{(n)}$, $n = 1 \ldots \mathcal{K}$ for body part i, we draw samples from the importance function

$$\mathbf{I}_i(\mathbf{x}_i) \sim \sum_{n=1}^{\mathcal{K}} \frac{1}{\mathcal{K}} \mathcal{N}(\mathbf{x}_i | \mathbf{Ix}_i^{(n)}, \mathbf{\Sigma}_i), i \in \mathcal{S} \setminus \{h, t\}, \tag{5.12}$$

where $\mathbf{\Sigma}_i$ is a diagonal covariance matrix. Note that a small number of $\mathcal{K}$ good hypotheses facilitate efficient sampling and inference process although it may have adverse effects if the value is too small.

5.4.2.3. Importance function for torso pose. Locating the torso region may be the most important task in human pose estimation since it is connected to most of the other body parts. However, detecting a torso part is difficult as it is usually clothed and thereby has a large variation in appearance. Without salient image cues (e.g., color and texture) to facilitate the detection process, we utilize line segments extracted from the probabilistic Hough transform [**66**] to assemble good shape hypotheses of the torso part.

A Canny edge detector is first applied to build the edge map, and then a probabilistic Hough transform is performed to detect those near-horizontal and near-vertical line segments. For each combination of a pair of vertical line segments, l_{v1}, l_{v2} and a pair of horizontal line segments l_{h1}, l_{h2}, let their corner points of the assembled shape be $p_{v1,h1}$, $p_{v1,h2}$, $p_{v2,h1}$, and

$p_{v2,h2}$ respectively. Torso hypotheses are obtained by solving an optimization problem with an objective function specified by

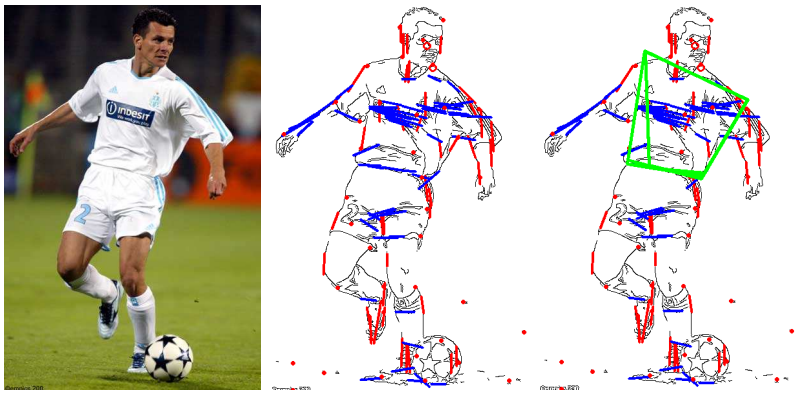
(1) The normalized shape of a good hypothesis should be reconstructed by the learned PCA subspace of the torso with minimum error.
(2) The distance between a good hypothesized torso part should be as close to the detected face as possible.
(3) The two vertical lines, l_{v1}, l_{v2} should be as symmetric as possible in the assembled shape.

subject to the constraints that $p_{v1,h1}$, $p_{v1,h2}$, $p_{v2,h1}$, and $p_{v2,h2}$ are within the range of image.

For each of the $\mathcal{M}$ torso hypotheses $\mathbf{Ix}_t^{(n)}$ obtained by solving the above-mentioned optimization problem ($n = 1, \ldots, \mathcal{M}$ and usually $\mathcal{M} < 10$), we compute the response of edges extracted by the Canny detector with likelihood $\beta_t^{(n)}$ using functions similar to Eq. 5.4 and Eq. 5.5. The importance sampling function for the torso pose is specified by a Gaussian mixture, i.e.,

$$\mathbf{I}_t(\mathbf{x}_t) \sim \sum_{n=1}^{\mathcal{M}} \beta_t^{(n)} \mathcal{N}(\mathbf{x}_t | \mathbf{Ix}_t^{(n)}, \mathbf{\Sigma}_t). \tag{5.13}$$

where $\mathbf{\Sigma}_t$ is the diagonal covariance matrix. Fig. 5.8 shows one example of the detected near-horizontal and near-vertical line segments from the probabilistic Hough transform, and the corresponding torso hypotheses. Although the number of combinations using horizontal and vertical lines is large, solving the above-mentioned optimization problem significantly prunes the number of torso hypotheses (i.e, $\mathcal{M} < 10$), thereby facilitating efficient and effective inference.



(a) (b) (c)

Figure 5.8. (a). Original image. (b). Line segments extracted by probabilistic Hough transform (red for near-vertical and blue for near-horizontal lines). (c). Torso hypotheses assembled from the line segments shown in (b).

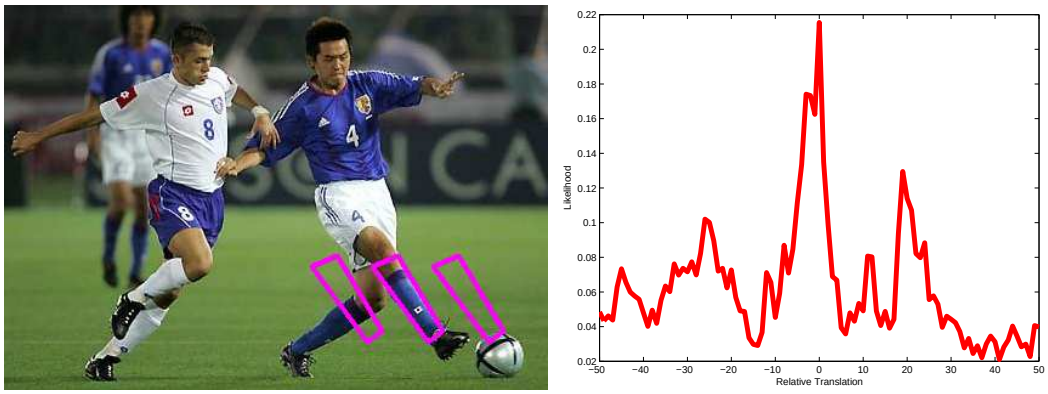
5.5. Experiments on human pose estimation

For concreteness, we apply our algorithm to estimate pose of soccer players in images. The proposed algorithm can be extended to estimate human pose in other domains.

5.5.1. Validation of the likelihood model

To demonstrate the effectiveness of the likelihood function proposed in Sec. 5.3.3, we generate a number of *left-lower-leg* hypotheses by translating the correctly labeled body part horizontally as shown in Fig. 5.9(a), and their likelihoods are shown in Fig. 5.9(b).

As exemplified in Fig. 5.9(b) the maximum likelihood occurs at the correct labeled location (i.e., 0 translation horizontally). The two small peaks correspond to the cases when one of the *left* and *right* lines of the shape pose is aligned with the boundary of the *left-lower-leg* in the image. The likelihood plots for the other body parts are similar to Fig. 5.9(b) except the likelihood model for the torso may not peak at the correct labeled location and may have



(a) (b)

Figure 5.9. (a) Translation of the *left-lower-leg* part with respect to the correct location horizontally. (b) Likelihoods of the translated *left-lower-leg* hypotheses from the correct location.

more local peaks (due to noisy edge response). This observation indicates that the difficulty of constructing a likelihood model of the torso part using only edge cues.

5.5.2. Pose estimation results

To learn the PCA subspace for each body part, we collected a set of 50 images and manually labeled the quadrangular shapes and poses of human body parts which best match human perception. For pose estimation experiments, we gathered another set of 30 images and manually located the body parts as ground truth (We will make the test image set publicly available at appropriate time.). These images contain humans with large variation in pose and backgrounds, as well as occlusions either due to clothing or view angles. The values of the diagonal covariance matrices in importance functions Eq. 5.11-Eq. 5.13 are empirically learned from the training image set.

Empirical results on estimating pose of soccer players in single images are illustrated in Fig. 5.10 where the best estimated shapes and locations of body parts are enclosed with quadrangles (video demo is available upon request.). The experimental results show that

Figure 5.10. Experimental results of human pose estimation.

		Head	Torso	LUA	LLA	RUA
RMSE		14.32	18.96	14.62	11.85	19.52
	RLA	LUL	LLL	RUL	RLL	Overall
RMSE	19.01	23.75	18.19	20,48	18.98	17.96

Table 5.1. Average root mean square error (RMSE) of the estimated 2-D pose for each body part and for the whole body (e.g., LUA refers to left-upper-arm).

our algorithm is able to locate the body parts and estimate their pose well even though they appear in different posture, background, view angles and lighting conditions. Our algorithm is able to infer poses which are heavily occluded in Fig. 5.10(f)-(g) as a result of data driven importance sampling from the visual cues. Specifically, the left lower leg of the player in Fig. 5.10(f) is located as a result of the best pose estimation using image likelihoods and importance function Eq. 5.12. Similarly, the occluded body parts and their poses in Fig. 5.10(h)-(j) are inferred using the proposed DDBPMC algorithm.

We evaluate the accuracy of our pose estimation algorithm by computing the root mean square errors (RMSE) between the estimated body pose enclosed by quadrangles and the ground truth, i.e., the RMSE between the four corner points of the two quadrangles. The average RMSE of each body part as well as that of the overall full body pose estimation over the 30 test images are presented in Table 5.1. At first glance, it seems that the RMSE of our algorithm is larger than the result of 20 test images reported in [**69**] even though the test sets are different. Nevertheless, we compute the accuracy of four points for each body parts while they just evaluated the accuracy of the joint locations, and thus the RMSE comparison is not justified. Further, the number of points set we compute is larger than that in [**69**]. Another complicating factor is the difficulty of determining what the "ground truth" of body pose is, as a result of covered clothing and difference of human perception in labeling body parts as well as pose precisely. Finally, the average RMSE of each image

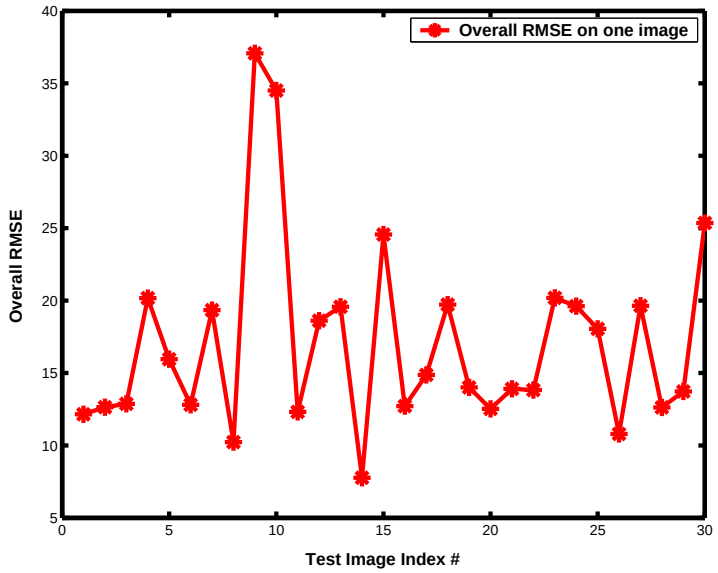


Figure 5.11. Overall RMSE of each of the test images.

is presented in Fig. 5.11 to show the distribution of the overall RMSE among the 30 test images.

The current implementation of the proposed algorithm draws 500 samples for each of the body parts, and the message passing process of the DDBPMC algorithm is iterated 6 times. Without code optimization, it takes about 2 to 3 minutes to process an image on a Pentium IV 1.7 GHz machine with 256 MB memory.

5.5.3. Discussions

Compared with the most relevant work [**69**], the problem we address in this chapter is well posed rather than inferring 3-D pose from single 2-D images. Furthermore, the test images in our work are more challenging since they contain complex poses with occlusions in textured background. Finally, we have done a larger scale experiment.

Although the experimental results demonstrate success of our algorithm in pose estimation from single images, there are a few research issues to be explored. The body postures such as torso may be more accurately estimated with more complicated body shapes. However, the inference problem will be more complicated due to the increasing degrees of freedom in body shape. The proposed algorithm sometimes fails when long line segments are observed near the torso region. This is not surprising since long line segments often cause problems in generating good hypotheses of the torso region.

5.6. Concluding remarks

We propose a rigorous statistical formulation for 2-D human pose estimation from single images. The theoretic foundation of this work is based on a Markov network, and the estimation problem is to infer pose parameters from observable cues such as appearance, shape, edge, and color. A novel data driven belief propagation Monte Carlo (DDBPMC) algorithm, which combines both top-down and bottom-up reasoning within a rigorous statistical framework, is proposed for efficient Bayesian inference. This is in contrast to the data driven Markov chain Monte Carlo (DDMCMC) algorithm in that DDBPMC carries out the Bayesian inference in parallel while the DDMCMC algorithm performs sequentially. Experimental results demonstrate the potency and effectiveness of the proposed method in estimating human pose from single images.

The proposed algorithm can be easily extended to better estimate human pose in situations where contour or motion cues abound. Our future work will focus on integrating visual cues to build better data driven importance functions for a more efficient pose estimation algorithm.

CHAPTER 6

Robust integration of inconsistent measurement

Another challenging issue in the proposed collaborative measurement integration framework is that the different visual measurements may be inconsistent. When inconsistency happens, it is obvious that there exists false measurements. It is harmful to blindly integrate all the measurements without excluding the false ones. This chapter presents our preliminary theoretic study on this complication, which results in a robust measurement integration framework based on a probabilistic variational Bayesian method. Encouraging results are obtained when we apply it to the task of robust part based ensemble tracking.

6.1. Introduction

In many vision problems, estimations are made based on integrating multiple sources of measurements to reduce the uncertainty. A measurement can generally be characterized by a mean vector and a covariance reflecting its uncertainty (multi-modal measurement can be treated as multiple measurements). To list a few examples, different sources can be different visual cues such as color and contour [**130**], different limbs of an articulated body [**105**, **126**], different components of one object [**37**, **50**], neighborhoods of a pixel in motion estimation [**16**], and dynamics and image observations in visual tracking [**52**]. This is a fundamental problem.

Most existing integration methods assume the consistency among various sources [**61**, **72**]. If the different sources are independent and consistent, the optimal integration can

be obtained by the best linear unbiased estimator (BLUE) [**72**]. If they are correlated but consistent, the covariance intersection (CI) [**61**] obtains a consistent and conservative estimate. However, the consistency assumption may not hold in practice. Basically, if two measurements can be regarded as being generated from the same model (e.g., a Gaussian), then they can be treated consistent, otherwise they are inconsistent. The measurements from different sources can be very confident (i.e., small covariance) but are quite different. They do not agree with each other and it makes less sense to fuse them together forcefully. Measurement inconsistency fails both BLUE and CI methods.

Indeed, this problem is not uncommon in computer vision applications. For example, a wrong dynamic prediction in Bayesian visual tracking is very likely to be inconsistent with detected image observations. This is especially true when the target presents sudden dynamic changes. Such an inconsistency shall fail Kalman filtering that is based on BLUE. In part-based tracking, the measurements of different parts may be conflicting when some parts are distracted by camouflages or occluded. The aperture problem in motion estimation is another example [**16**].

Unfortunately, the handling of inconsistency is not well addressed in the literature. Therefore, it is desirable to carry out some basic study of inconsistency in order to identify the solution to robust measurement integration. We are particularly interested in answering two questions: (a) how can we detect inconsistency from the measurements? and (b) how can we handle it in integration? We need to develop principled criteria to characterize inconsistency and develop efficient method to detect and resolve it.

This chapter describes a novel *distributed* integration approach based on the theory of Markov networks [**44**]. Although Markov networks were widely applied to solve visual inference problems [**29**, **105**, **126**], the study of information fusion of the inference over Markov

networks is largely remained unexplored. In this chapter, we proved a new theorem that provides two algebraic criteria to examine the *consistency* and *inconsistency* for pair-wise measurements. In addition a general criterion is proposed to detect inconsistency in a general setting.

Since the presence of inconsistency implies the presence of false or outlier measurements, our method can automatically identify the inconsistent measurements and eliminate the false ones for further integration. Based on the proposed integration approach, we have developed a robust part-based tracking algorithm in which measurements of various parts are integrated for tracking, although there exists some inconsistent ones.

There are some previous work which are aware of the *inconsistency* problem such as the covariance union (CU) [**117**] and the variable bandwidth density fusion (VBDF) [**16**]. They either increase the covariance of the integrated estimate to achieve covariance consistency with each of the integrated measurements [**117**], or seek for the most salient mode across all scales of the measurements kernel density [**16**]. None of them provides a principled criteria to evaluate measurement inconsistency, i.e., they can not determine when two measurements can be regarded as being obtained from one model.

The distributed probabilistic model for measurement integration is presented in Sec. 6.7. We then present our theoretical study of measurement inconsistency for pairwise measurements in Sec. 6.3, which provides two algebraic conditions to determine measurement consistency and inconsistency. Based on our theoretical study of inconsistency, we propose a general criterion for false measurement arbitration based on majority rule in Sec. 6.4. After that, we present the proposed robust measurement integration framework in Sec. 6.5. Extensive experimental results are presented in Sec. 6.6, followed by our conclusion remarks in Sec. 6.7.

6.2. Formulation of multi-source integration

Markov network provides a principled methodology for the *distributed* integration of multiple sources. The joint posterior defined on a Markov network is

$$P(\mathbf{X}|\mathbf{Z}) = \frac{1}{C} \prod_{\{i,j\}\in\mathcal{E}} \psi_{ij}(\mathbf{x}_i, \mathbf{x}_j) \prod_{i\in\mathcal{V}} \phi_i(\mathbf{x}_i, \mathbf{z}_i), \tag{6.1}$$

where C is a normalization constant, $\mathbf{X} = \{\mathbf{x}_i : i = 1 \dots N\}$, $\mathbf{Z} = \{\mathbf{z}_i : i = 1 \dots N\}$ and N is the number of sources modeled in the Markov network.

Each $\mathbf{x}_i$ denotes the integrated estimate at node i, and $\mathbf{z}_i$ is the local measurement of source i. Set $\mathcal{V}$ indicates the set of $\{\mathbf{x}_i, \mathbf{z}_i\}$ pairs and each pair has a compatibility function $\phi_i(\mathbf{x}_i, \mathbf{z}_i)$. Let $\mathbf{x}_i$, $\mathbf{z}_i$ be in $\mathcal{R}^n$, since the measurement is a $\{\mathbf{z}_i, \mathbf{\Sigma}_i\}$ pair, $\phi_i(\mathbf{x}_i, \mathbf{z}_i)$ is in nature a Gaussian, i.e.,

$$\phi_i(\mathbf{x}_i, \mathbf{z}_i) = \frac{1}{\sqrt{(2\pi)^n |\mathbf{\Sigma}_i|}} e^{\left(-\frac{1}{2}(\mathbf{z}_i - \mathbf{x}_i)^T \mathbf{\Sigma}_i^{-1} (\mathbf{z}_i - \mathbf{x}_i)\right)}. \tag{6.2}$$

Set $\mathcal{E}$ defines the neighborhood relationships in the Markov network. If $\mathbf{x}_j$ is the neighbor of $\mathbf{x}_i$, then $\mathbf{x}_j$ can provide a predictive estimate $f_{ij}(\mathbf{x}_j)$ for $\mathbf{x}_i$. $\psi_{ij}(\mathbf{x}_i, \mathbf{x}_j)$ is the compatibility function of the neighboring $\mathbf{x}_i$ and $\mathbf{x}_j$, i.e., a Gaussian

$$\psi_{ij}(\mathbf{x}_i, \mathbf{x}_j) = \frac{e^{\left\{-\frac{(\mathbf{x}_i - f_{ij}(\mathbf{x}_j))^T (\mathbf{x}_i - f_{ij}(\mathbf{x}_j))}{2\sigma_{ij}^2}\right\}}}{\sqrt{(2\pi)^n}\sigma_{ij}^n} \tag{6.3}$$

$$\doteq \frac{e^{\left\{-\frac{(\mathbf{x}_i - \mathbf{A}_{ij}\mathbf{x}_j - \mu_{ij})^T (\mathbf{x}_i - \mathbf{A}_{ij}\mathbf{x}_j - \mu_{ij})}{2\sigma_{ij}^2}\right\}}}{\sqrt{(2\pi)^n}\sigma_{ij}^n}, \tag{6.4}$$

which indicates if $\mathbf{x}_i$ and $f_{ij}(\mathbf{x}_j)$ can be regarded as being drawn from one common model and σ_{ij}^2 is the scalar variance. When f_{ij} is nonlinear, we linearize it by Taylor expansion, i.e., $\mu_{ij} = f_{ij}(\mathbf{0})$ and $\mathbf{A}_{ij} = \frac{\partial f_{ij}(\mathbf{x}_j)}{\partial \mathbf{x}_j}|_{\mathbf{x}_j = \mathbf{0}}$ is the $n \times n$ Jacobian. So we only consider the setting

of Eq. 6.4. The σ_{ij}^2 indeed models the uncertainties between the local estimate $\mathbf{x}_i$ and the neighborhood estimate $\mathbf{A}_{ij}\mathbf{x}_j + \mu_{ij}$.

The integration of all the measurements is to perform the Bayesian inference on Eq. 6.1. Nevertheless, when some measurements are inconsistent with the others, it indicates there are false ones. Blindly integrating them will jeopardize the whole integration process. Let $\mathbf{O} = \{\mathbf{O}_i, i = 1 \dots N\}$ be the binary set to indicate if $\mathbf{z}_i$ is false, i.e., $\mathbf{O}_i = 1$ means it is and vice versa. $\mathbf{O}$ divides $\mathbf{Z}$ into two sets, i.e., the false set $\mathbf{Z}_{\mathcal{O}}$ and the normal set $\mathbf{Z}_{\bar{\mathcal{O}}} = \mathbf{Z} \setminus \mathbf{Z}_{\mathcal{O}}$. Reliable integration requires eliminating the false ones, i.e., we should perform the Bayesian inference on

$$P(\mathbf{X}|\mathbf{Z}_{\bar{\mathcal{O}}}) = \frac{1}{C'} \prod_{\{i,j\}\in\mathcal{E}} \psi_{ij}(\mathbf{x}_i, \mathbf{x}_j) \prod_{\mathbf{z}_i \in \mathbf{Z}_{\bar{\mathcal{O}}}} \phi_i(\mathbf{x}_i, \mathbf{z}_i), \tag{6.5}$$

where C' is again for normalization. Before we can achieve that, we need a rigorous criteria to judge *inconsistency*. For integration, this concept is always qualitative [**117**], we proceed to provide principled quantitative criteria.

6.3. Measurements inconsistency

Intuitively, assume $\mathbf{A}_{ij}$ and μ_{ij} be known, given all the $\{\mathbf{z}_i, \mathbf{\Sigma}_i\}$, the estimate of σ_{ij}^2 is a natural indicator of whether $\mathbf{x}_i$ and $\mathbf{A}_{ij}\mathbf{x}_j + \mu_{ij}$ is consensus, i.e., if σ_{ij}^2 is very small, then they are consensus since $\psi_{ij}(\mathbf{x}_i, \mathbf{x}_j)$ is approaching to a delta function, and vice versa. Denote $\mathbf{\Theta} = \{\sigma_{ij}^2 : \{i,j\} \in \mathcal{E}\}$, Eq. 6.1 is indeed $P(\mathbf{X}|\mathbf{\Theta}, \mathbf{Z})$. The MAP estimate of $\mathbf{x}_i$ and the ML estimate of $\mathbf{\Theta}$ can be obtained by the following Bayesian EM algorithm [**85**], i.e.,

$$\mathbf{x}_i = (\mathbf{\Sigma}_i^{-1} + \sum_{j\in\mathcal{N}(i)} \frac{1}{\sigma_{ij}^2}\mathbf{I})^{-1} \left(\mathbf{\Sigma}_i^{-1}\mathbf{z}_i + \sum_{j\in\mathcal{N}(i)} \frac{1}{\sigma_{ij}^2}(\mathbf{A}_{ij}\mathbf{x}_j + \mu_{ij}) \right) \tag{6.6}$$

$$\sigma_{ij}^2 = \frac{1}{n}(\mathbf{x}_i - \mathbf{A}_{ij}\mathbf{x}_j - \mu_{ij})^T(\mathbf{x}_i - \mathbf{A}_{ij}\mathbf{x}_j - \mu_{ij}) \tag{6.7}$$

Fixing $\mathbf{\Theta}$, the E-Step in Eq. 6.6 obtains the MAP estimate of $\mathbf{x}_i$ by fixed-point iteration. It is actually performing the BLUE [**72**] fusion of the local estimate and neighborhood estimate. Fixing $\mathbf{X}$, the M-Step in Eq. 6.7 maximizes $P(\mathbf{X}|\mathbf{\Theta}, \mathbf{Z})$ w.r.t. $\mathbf{\Theta}$. Combining the two steps together also constitutes a fixed-point iteration for σ_{ij}^2. In practice, we add a small regularization constant ϵ (e.g., 0.01) on the right-side of Eq. 6.7 to avoid the numerical problem of zero.

Another intuition is that the consensus between the estimate of $\mathbf{x}_i$ and $\mathbf{A}_{ij}\mathbf{x}_j + \mu_{ij}$ is equivalent to the consistency of the measurements $\{\mathbf{z}_i, \mathbf{\Sigma}_i\}$ and $\{\mathbf{z}_j, \mathbf{\Sigma}_j\}$. Therefore, when $\mathbf{z}_i$ and $\mathbf{z}_j$ are consistent, the estimate of $\mathbf{x}_i$ and $\mathbf{A}_{ij}\mathbf{x}_j + \mu_{ij}$ will be consensus, i.e., they will be almost the same. From Eq. 6.7, the estimate of σ_{ij}^2 will always approach to zero, i.e., zero is the only fixed-point. On the contrary, if they are inconsistent, then the estimate of $\mathbf{x}_i$ and $\mathbf{A}_{ij}\mathbf{x}_j + \mu_{ij}$ may deviate from each other, i.e., the convergent results of σ_{ij}^2 may be non-zero. This indicates that there exists non-zero fixed-point for σ_{ij}^2. These motivate us for the following definition for inconsistency.

Definition 6.3.1. *If zero is the only fixed-point for σ_{ij}^2 in the Bayesian EM, $\{\mathbf{z}_i, \mathbf{\Sigma}_i\}$ and $\{\mathbf{z}_j, \mathbf{\Sigma}_j\}$ are* consistent*; if there exists non-zero fixed-points for σ_{ij}^2, they are* inconsistent.

This definition motivates us to detect the inconsistency by checking the convergent value of σ_{ij}^2. We thus have the following criterion to test consistency.

Criterion 6.3.2. *With a proper initialization, if the convergent results of σ_{ij}^2 in the Bayesian EM approaches to zero, then $\{\mathbf{z}_i, \mathbf{\Sigma}_i\}$ and $\{\mathbf{z}_j, \mathbf{\Sigma}_j\}$ are consistent. If it converges to a non-zero value, then they are inconsistent.*

In practice, a *proper* initialization should guarantee σ_{ij}^2 to converge to a non-zero fixed-point if there exists one. Such a condition is necessary because zero is always a trivial fixed-point (see App. C). For better mathematical understanding of Definition 6.3.1, we proved the following Theorem 6.3.3 by studying the convergence of the Bayesian EM for pair-wise measurements. In Corollary 6.3.4, we also present a guidance to choose the *proper* initialization for Criterion 6.3.2 .

Theorem 6.3.3. *For a Markov network which models the integration of two sources, denote* $\hat{\mathbf{z}}_2 = \mathbf{A}_{12}\mathbf{z}_2 + \mu_{12}$, $\hat{\mathbf{\Sigma}}_2 = \mathbf{A}_{12}\mathbf{\Sigma}_2\mathbf{A}_{12}^T$, $\mathbf{P} = \mathbf{\Sigma}_1 + \hat{\mathbf{\Sigma}}_2$ *which is* real positive definite*, C_p the 2-norm conditional number and σ_{Pmax}^2 the largest eigenvalue of $\mathbf{P}$, and $\hat{\sigma}_{12}^2$ as the convergent results of σ_{12}^2 in the Bayesian EM. We have*

(a) *There exists a zero and at least one non-zero $\hat{\sigma}_{12}^2$ if*

$$\frac{1}{n}(\mathbf{z}_1 - \hat{\mathbf{z}}_2)^T\mathbf{P}^{-1}(\mathbf{z}_1 - \hat{\mathbf{z}}_2) \geq 2 + \sqrt{C_p} + \frac{1}{\sqrt{C_p}}. \tag{6.8}$$

(b) *$\hat{\sigma}_{12}^2$ can only be zero if*

$$\frac{1}{n}(\mathbf{z}_1 - \hat{\mathbf{z}}_2)^T\mathbf{P}^{-1}(\mathbf{z}_1 - \hat{\mathbf{z}}_2) < 4. \tag{6.9}$$

(c) *When there exists non-zero $\hat{\sigma}_{12}^2$, at least one of them is such that $0 < \hat{\sigma}_{12}^2 \leq \sigma_{Pmax}^2$*

The proof is presented in App. C. Highlighted by Theorem 6.3.3(c), we have the following corollary.

Corollary 6.3.4. *Under the same condition of Theorem 6.3.3, initializing* σ_{12}^2 *to be the largest eigen-value* σ_{Pmax}^2 *or the trace* $T(\mathbf{P})$ *of* $\mathbf{P}$ *in the Bayesian EM can guarantee a non-zero convergence for* σ_{12}^2 *if there exists one.*

The proof is presented in App. D. Theorem 6.3.3 and Corollary 6.3.4 provide a sound mathematical justification of Definition 6.3.1 about inconsistency and consistency. We denote the left side of Eq. 6.8 and Eq. 6.9 as $d(\mathbf{z}_1, \mathbf{z}_2)$, which is in fact a Mahalanobis distance. In principle, when $d(\mathbf{z}_1, \mathbf{z}_2)$ is too large, statistically $\{\mathbf{z}_1, \mathbf{\Sigma}_1\}$ and $\{\mathbf{z}_2, \mathbf{\Sigma}_2\}$ are significantly deviated from each other and thus they are inconsistent. In this case there exists at least one non-zero convergence of σ_{12}^2. On the other hand, if $d(\mathbf{z}_1, \mathbf{z}_2)$ is small, statistically $\{\mathbf{z}_1, \mathbf{\Sigma}_1\}$ and $\{\mathbf{z}_2, \mathbf{\Sigma}_2\}$ are not deviated from each other and thus they are consistent. Then there will be only zero convergence for σ_{12}^2.

Theorem 6.3.3(a) and (b) present two algebraic criteria (sufficient conditions) to judge if $\{\mathbf{z}_1, \mathbf{\Sigma}_1\}$ and $\{\mathbf{z}_2, \mathbf{\Sigma}_2\}$ are inconsistent or consistent, i.e., if Eq. 6.8 holds, then they are inconsistent, and they are consistent if Eq. 6.9 holds. The following remarks would make the understanding more clear:

- Since $B_c = 2 + \sqrt{C_p} + \frac{1}{\sqrt{C_p}} \geq 4$, if $4 \leq d(\mathbf{z}_1, \mathbf{z}_2) < B_c$, we can not directly tell if there exists a non-zero $\hat{\sigma}_{12}^2$. In other words, we can not immediately decide the consistency unless we run the Bayesian EM.
- In one dimensional case, i.e., $n = 1$, we have $B_c = 4$. Then the inconsistency/consistency of $\mathbf{z}_1$ and $\mathbf{z}_2$ can be determined by testing if $d(\mathbf{z}_1, \mathbf{z}_2) \gtreqless 4$.
- For $n \geq 2$, if C_p is good to be near 1, then B_c would be very close to 4. The interval $[4, B_c)$ would be very tight. Then either B_c or 4 can be approximately used for detecting inconsistency similar to the case $n = 1$.

- For $n \geq 2$, if C_p is not good to be very large, then $B_c \gg 4$. We must run the Bayesian EM with a proper initialization to judge the consistency when $d(\mathbf{z}_1, \mathbf{z}_2)$ falls in $[4, B_c)$.
- In a general setting, from Corollary 6.3.4, the largest eigenvalue or the trace of $\mathbf{\Sigma}_i + \mathbf{\Sigma}_j + \sum_{k \in \mathcal{N}(i,j)} \mathbf{\Sigma}_k$ is a *proper* initialization, where $\mathcal{N}(i,j)$ is the neighborhood of i and j. The trace is preferable since it can be more efficiently obtained.

6.4. Detection of inconsistency and falseness

Based on Criterion 6.3.2, let L_{ij} be the binary variable to indicate whether $\{\mathbf{z}_i, \mathbf{\Sigma}_i\}$ and $\{\mathbf{z}_j, \mathbf{\Sigma}_j\}$ are inconsistent, i.e., $L_{ij} = 1$ represents that they are and vice versa. Then the criterion to identify the inconsistency is

$$L_{ij} = \begin{cases} 0 & if\ \sigma_{ij}^2 \leq \epsilon \\ 1 & if\ \sigma_{ij}^2 > \epsilon \end{cases}, \tag{6.10}$$

where ϵ is the same regularization constant added in Eq. 6.7.

After the detection of inconsistency, the majority rule is adopted to determine if $\{\mathbf{z}_i, \mathbf{\Sigma}_i\}$ is false, i.e., if $\{\mathbf{z}_i, \mathbf{\Sigma}_i\}$ is inconsistent with the majority of its neighbors, then it is false, and vice versa. Without any other knowledge, the majority rule may be the best one to discriminate false measurements. The basic assumptions are that there are at least three sources and the majority of the sources will obtain correct and thus consistent measurements. Since $\mathbf{O}_i$ is the binary variable to indicate if $\mathbf{z}_i$ is false, suppose part i has M_i neighbors, then

$$\mathbf{O}_i = \begin{cases} 0 & if\ \sum_{j \in \mathcal{N}(i)} L_{ij} \leq \lfloor \frac{M_i}{2} \rfloor \\ 1 & if\ \sum_{j \in \mathcal{N}(i)} L_{ij} > \lfloor \frac{M_i}{2} \rfloor \end{cases} \tag{6.11}$$

where $\lfloor \frac{M_i}{2} \rfloor$ is the largest integer that is not larger than $\frac{M_i}{2}$.

However, when the degrees (i.e., the number of neighboring nodes) of the nodes in the Markov network are highly unbalanced, the majority rule may fail even if there are less than 50% false measurements. One such example would be that the connections of $N > 6$ nodes form a circle and meanwhile the nodes $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$ are connected with all the other nodes. Then if the measurements $\mathbf{z}_1$, $\mathbf{z}_2$ and $\mathbf{z}_3$ are false and thus inconsistent with the others, all the other measurements will be regarded as "false" from Eq. 6.11.

Such a problem may not exist when the degrees of the nodes are well balanced. This reveals to us that in order to well exploit Eq. 6.11, we must construct a balanced Markov network to integrate the multiple sources, i.e., the degrees of the nodes must be close to one another.

6.5. Robust integration for ensemble tracking

Given all $\{\mathbf{z}_i, \mathbf{\Sigma}_i\}$s, we propose a two-stage robust integration approach:

(1) `False discrimination:` Perform the Bayesian EM on the original Markov network $\mathcal{M}_o$ defined by Eq. 6.1 and then identify the false measurements set $\mathbf{Z}_{\mathcal{O}}$ based on Eq. 6.11.

(2) `Robust Integration:` Remove all $\mathbf{z}_i \in \mathbf{Z}_{\mathcal{O}}$, from $\mathcal{M}_o$. This forms the reduced Markov network $\mathcal{M}_r$ defined by Eq. 6.5. Perform the Bayesian EM on $\mathcal{M}_r$ to obtain the estimates for all $\mathbf{x}_i$ with $\mathbf{Z}_{\mathcal{O}}$ being removed from Eq. 6.6 and Eq. 6.7.

It is a completely *distributed* robust integration approach, where all the operations are performed individually at each node of the Markov network. After the false measurement at one source node i has been eliminated, as we can observe from Eq. 6.6 (eliminating $\mathbf{z}_i$ and $\mathbf{\Sigma}_i$ from it), the estimate of $\mathbf{x}_i$ will rely purely on the neighborhood estimates.

It can be immediately applied to part-based visual tracking (i.e., an ensemble tracker), where $\psi_{ij}(\mathbf{x}_i, \mathbf{x}_j)$ captures the structured constraints between two neighboring parts. It is also general to incorporate different tracking algorithms to obtain the part measurements $\{\mathbf{z}_i, \mathbf{\Sigma}_i\}$, such as particle filtering [**52**] and flow based Lucas-Kanade tracker (LK) [**102**], etc..

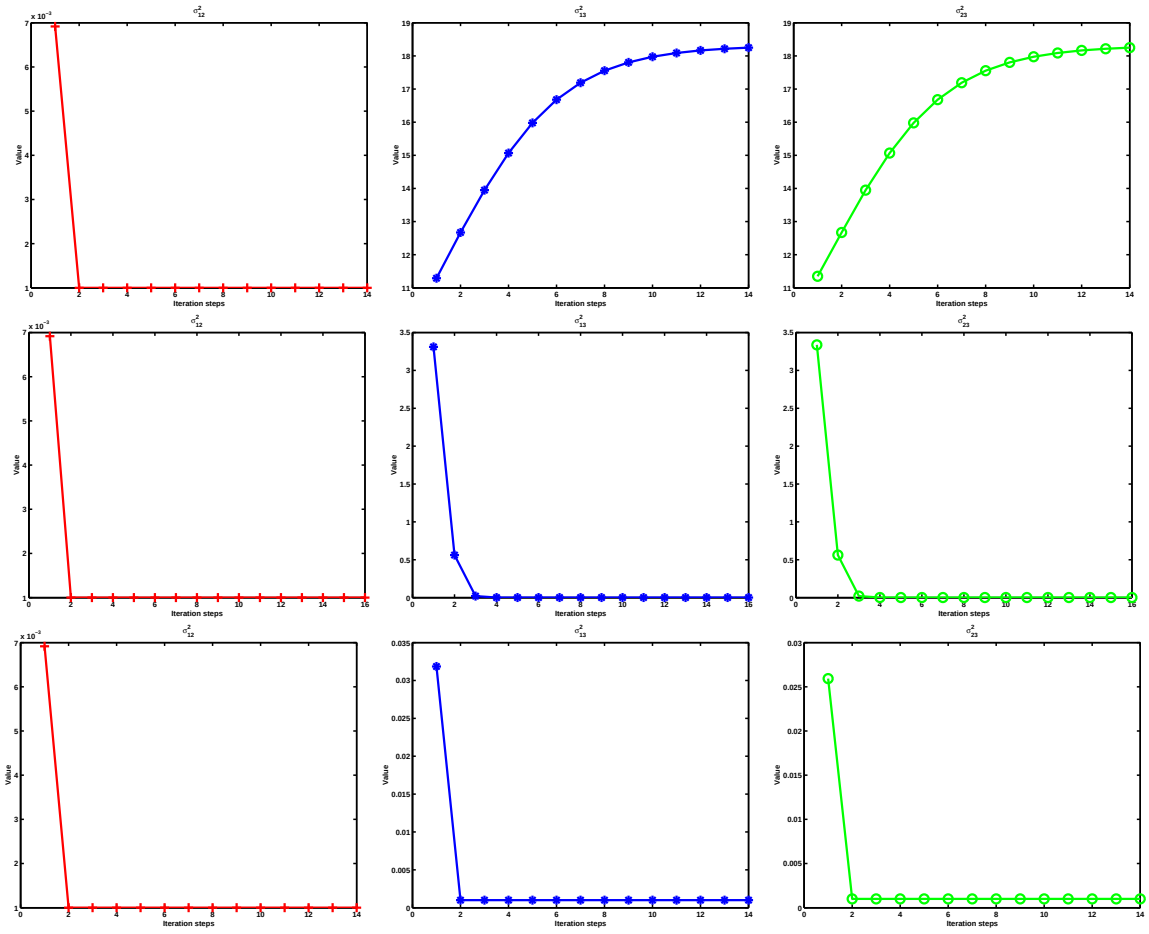
There are three situations: (1) The measurements of all the parts are normal and consistent. (2) The measurements of some parts are *missing*, i.e., the $\phi_i(\mathbf{x}_i, \mathbf{z}_i)$ is a Gaussian with large co-variance. This might happen when the visual pattern of the target undergoes large variations but the visual model does not capture it well. (3) The measurements of some parts are inconsistent with those of the other. This implies that some measurements are false and it may be caused by either occlusion, clutter or camouflage in visual tracking. Our robust integration approach handles all these three situations in a unified way.

6.6. Experiments

6.6.1. Illustrative numerical example

We adopt a $2D$ numerical example to demonstrate how σ_{ij}^2 changes during the Bayesian EM. The Markov network models three sources, which are neighbors of one another. Without loss of generality, we set all $\mathcal{A}_{ij} = \mathbf{I}$ and $\mu_{ij} = 0$. In all the simulations, we fix $\mathbf{z}_1 = [2.1, 2.2]^T$, $\mathbf{z}_2 = [2.2, 2.1]^T$ and $\mathbf{\Sigma}_1 = \mathbf{\Sigma}_2 = [2.0, 1.0; 1.0, 2.0]$. We then set $\{\mathbf{z}_3, \mathbf{\Sigma}_3\}$ to be different values to simulate the three situations. Highlighted by Corollary 6.3.4, we always initialize all σ_{ij}^2 to be the trace of $\mathbf{\Sigma}_1 + \mathbf{\Sigma}_2 + \mathbf{\Sigma}_3$.

We firstly simulate the case of false measurement, e.g., $\mathbf{z}_3 = [8.0, 9.0]^T$ and $\mathbf{\Sigma}_3 = [2.0, 1.0; 1.0, 2.0]$. It is obvious that $\{\mathbf{z}_3, \mathbf{\Sigma}_3\}$ is false. The changes of σ_{12}^2, σ_{13}^2 and σ_{23}^2 are presented in the first row of Fig. 6.1. As we can observe, σ_{12}^2 converges to 0.01, and both σ_{13}^2 and σ_{23}^2 converges to 18.25. Using Eq. 6.11, we easily identify $\mathbf{z}_3$ as a false measurement



(a). σ_{12}^2 (b). σ_{13}^2 (c). σ_{23}^2

Figure 6.1. The change of σ_{ij}^2 in the Bayesian EM. **First row**: measurement $\mathbf{z}_3$ is false. **Second row**: measurement $\mathbf{z}_3$ is missing. **Third row**: all the measurements are consistent.

and it will be eliminated in the robust inference step. The MAP estimates before false elimination are $\mathbf{x}_1 = [2.83, 2.87]^T$, $\mathbf{x}_2 = [2.83, 2.87]^T$ and $\mathbf{x}_3 = [6.65, 7.55]^T$, which are erroneous and can be rectified after we eliminated $\mathbf{z}_3$.

We then simulate the case of missing measurement, e.g., $\mathbf{z}_3 = [8.0, 9.0]^T$ with $\mathbf{\Sigma}_3 = [10.0, 1.0; 1.0, 10.0]$. Although $\mathbf{z}_3$ is deviated from $\mathbf{z}_1$ and $\mathbf{z}_2$, its covariance $\mathbf{\Sigma}_3$ is pretty large so it is still consistent with the others. The changes of σ_{ij}^2 are presented in the second row of Fig. 6.1. We can observe that all of them converge to 0.01. In fact, the MAP estimates

(a). #16 (b). #66 (c). #88 (d). #90 (e). #102 (f). #184

Figure 6.2. Results with flow measurement: the red, green, and yellow color denote false, normal and missing measurements, respectively.

(a). LK #90 (b). Holistic PF #108

Figure 6.3. Typical tracking failure (a). LK tracking frame #90. (b). Particle filtering with holistic appearance model frame #108.

are $[2.89, 2.94]^T$ for all $\mathbf{x}_i$. We can see $\mathbf{z}_3$ has been counted far less than the other two measurements and the bias has largely been rectified in the estimates.

Last we simulate the easiest case where all the measurements are reliable and consistent, e.g., $\mathbf{z}_3 = [1.9, 1.8]^T$ with $\mathbf{\Sigma}_3 = [2.0, 1.0; 1.0, 2.0]$. The change of σ_{ij}^2 is presented in the third row of Fig. 6.1. Again, they all converge to 0.01 as expected. The final MAP estimates are $[2.07, 2.03]^T$ for all $\mathbf{x}_i$. We have extensively run the simulations with different settings. The results are coherent with what are presented.

6.6.2. Robust part based ensemble tracking

6.6.2.1. Part based tracking with LK tracker. We first present the results using LK tracker [**102**] to obtain the part measurements. The test video clip is from the comedy cartoon "Tom and Jerry". The target is the poor cat Tom's face. Those "good features" [**102**]

in Tom's face region are detected to be the node of the Markov network. The face region is manually cropped as a rectangle in the first frame.

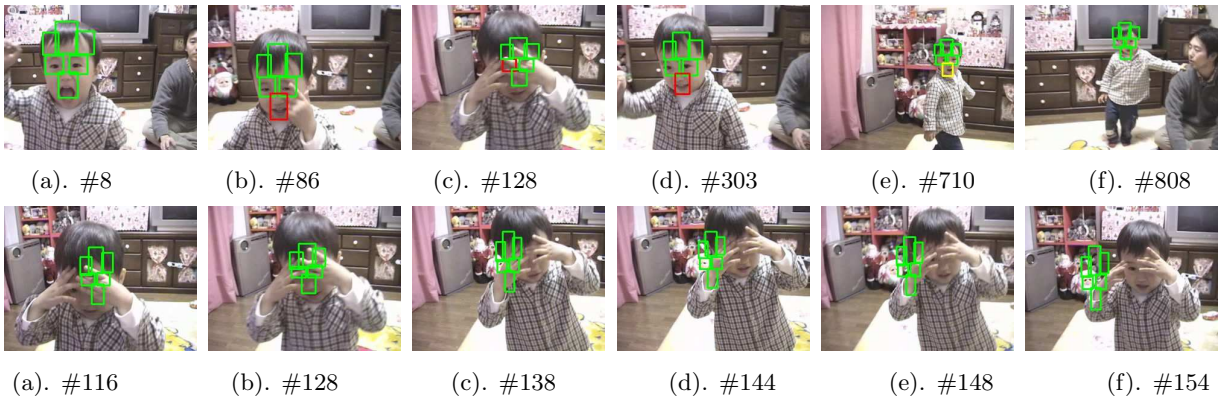
Each node is associated with a 7×7 image patch (the appearance model) centering at the feature point, and it is connected with the three nearest nodes. The $\mathbf{x}_i$ is the $2D$ position of the ith good feature. At the current frame, we set $\mathbf{A}_{ij} = \mathbf{I}_2$ and set μ_{ij} to be the relative position of part i and j in the previous frame. Each $\mathbf{z}_i$ is obtained by the flow based LK tracker. The $\mathbf{\Sigma}_i$ is obtained by evaluating the response distribution using SSD similar to that in [**141**].

We show some sample results in Fig. 6.2[1]. Our algorithm successfully identifies the false, missing and normal measurements, as shown in red, yellow and green, respectively. The video has 187 frames and our algorithm obtains robust results. With 50 parts, it runs at 10 frames/second without code optimization.

The pure LK tracker and the particle filtering (PF) with a holistic appearance model are easy to fail in this video clip. We show the typical failure cases in Fig. 6.3. The failures are due to the dramatic expression change (Fig. 6.3(a)), the sudden view changes and abrupt motion of Tom (Fig. 6.3(b)). The number of particles for holistic PF is 200 and all algorithms are initialized with the same rectangle.

6.6.2.2. Part based tracking with particle filtering. In this section, we present the tracking results using particle filtering [**52**] to obtain the part measurement. The $\mathbf{x}_i$ is four dimensional (two for translations and two for scalings). The target parts are selected manually and a fully connected Markov network is adopted. The $\mathbf{A}_{ij}$ and μ_{ij} are estimated from some manually annotated images by least square fitting. There is a residue error σ_{ij0}^2 from the least square fitting. It is used as the σ_{ij}^2 in the robust integration step, i.e., after

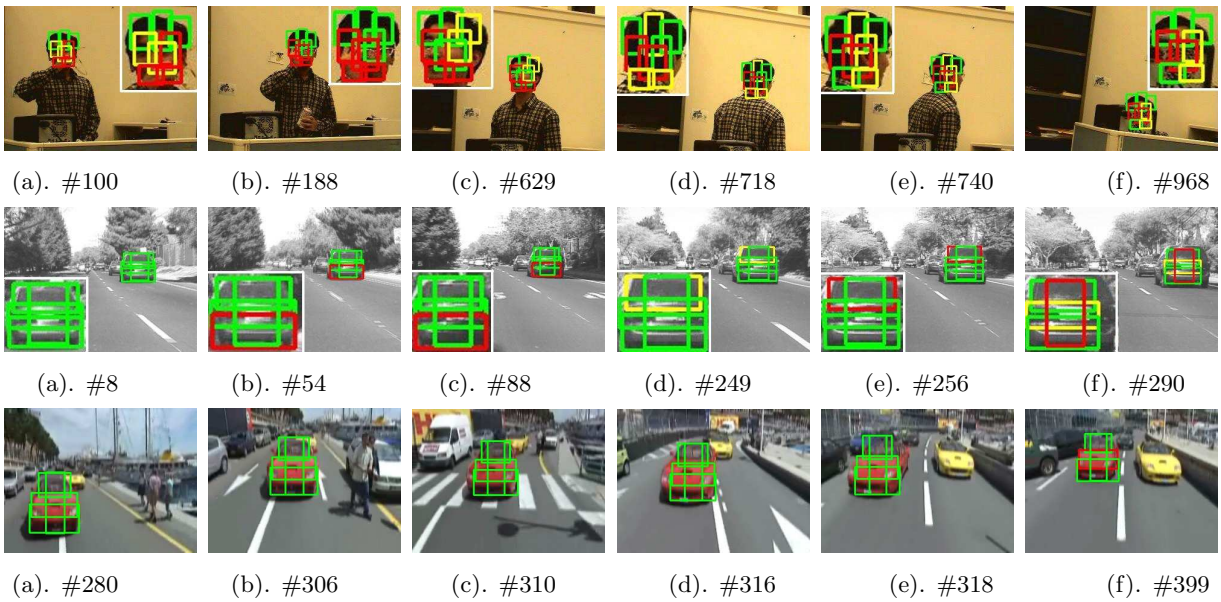
[1]All video results are available upon request



(a). #8 (b). #86 (c). #128 (d). #303 (e). #710 (f). #808

(a). #116 (b). #128 (c). #138 (d). #144 (e). #148 (f). #154

Figure 6.4. Comparison of robust integration by the proposed approach and blind integration without inconsistency detection and false elimination – **First row**: Proposed integrating approach (green-normal, red-false, yellow-missing). **Second row**: Blind integrating.



(a). #100 (b). #188 (c). #629 (d). #718 (e). #740 (f). #968

(a). #8 (b). #54 (c). #88 (d). #249 (e). #256 (f). #290

(a). #280 (b). #306 (c). #310 (d). #316 (e). #318 (f). #399

Figure 6.5. Results with PF measurement: Results in the first and second row are enlarged for better visual quality (green-normal, red-false, yellow-missing).

removing the false measurements, we fix $\sigma_{ij}^2 = \sigma_{ij0}^2$ and perform the Bayesian inference using Eq. 6.6. Note each component has a template image patch to build the appearance based likelihood model $\phi_i(\mathbf{x}_i, \mathbf{z}_i)$. The mean estimates and the covariances of the posterior particle sets are adopted as the part measurements $\{\mathbf{z}_i, \mathbf{\Sigma}_i\}s$.

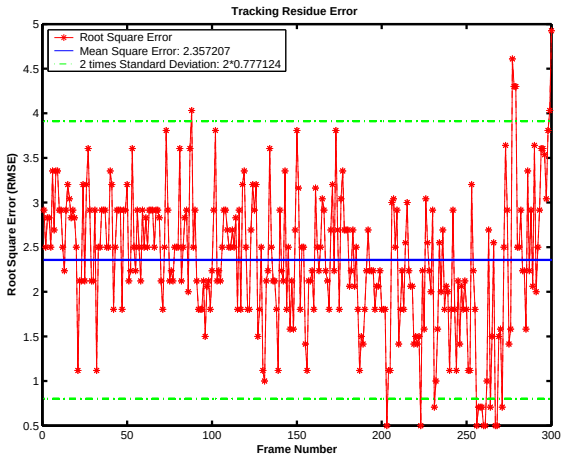
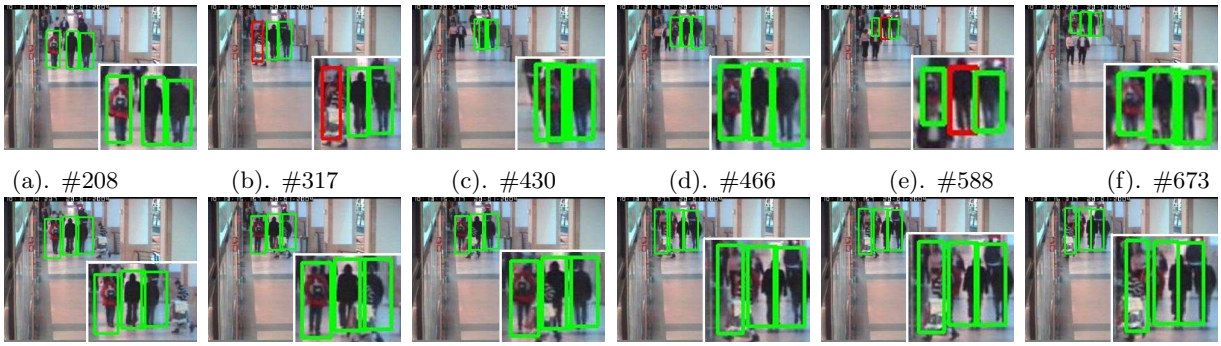


Figure 6.6. Root square errors of the results on the car sequences in the first row of Fig. 6.5.

We present sample results on different video sequences in Fig. 6.4 and Fig. 6.5. These test videos are typical, where the targets present large appearance variations due to the significant view, scale, lighting changes and the presence of occlusions. Fig. 6.4 shows the results of tracking the face of a kid. The first row of Fig. 6.4 shows the results of the proposed approach, where inconsistent measurements are detected and those false ones are eliminated. For comparison, the second row of Fig. 6.4 shows the results of blind integration without inconsistency detection and false elimination. Note how the tracking results have been distracted due to the integration of those false measurements during occlusion. The video has 820 frames.

In Fig. 6.5, we present sample image results on three other video sequences both including tracking faces and cars. From top to bottom, The videos have 1121, 348 and 399 frames, respectively. We also test the accuracy of the results shown in the second row of Fig. 6.5 on the two translation parameters. 300 frames are labeled and the centroid points of the labeled rectangle is adopted as the ground truth. For the tracking results, the centroid point of all the part rectangles is used as the overall translation parameters. We then calculate the root



(a). #208 (b). #317 (c). #430 (d). #466 (e). #588 (f). #673

(a). #280 (b). #306 (c). #310 (d). #316 (e). #318 (f). #322

Figure 6.7. Tracking a group of persons: **First row**: Our integrating approach (green-normal, red-false, yellow-missing). **Second row**: Blind integration. The results are enlarged for better visual quality.

(a). #8 (b). #28 (c). #88 (d). #168 (e). #188 (f). #228

Figure 6.8. Tracking a group of three cars by the proposed approach.

square error at each frame, as shown in Fig. 6.6. The root mean square error is 2.36 pixels with stand deviation 0.78 on 320×240 images. This shows the accuracy of the proposed approach.

6.6.2.3. Tracking a group of objects. A direct generalization of part based tracking is to track several objects moving in a group. We test the robust integration on part of a video sequence[2] where three persons walking in a corridor of a shopping mall. Again, a fully connected Markov network is adopted and the measurement of each person is obtained by particle filter. Some sample results are presented in the first row of Fig. 6.7. In Fig. 6.7(b) (first row), frame #317, the left person has been occluded by another person and the measurement is false. Our algorithm clearly identifies and corrects it. For comparison, we also

[2]From the EC Funded CAVIAR project/IST 2001 37540, URL: http://homepages.inf.ed.ac.uk/rbf/CAVIAR/

present the results by blind integration in the second row of Fig. 6.7. Note how it fails due to occlusion. We also present the tracking results by the proposed approach on a video sequence of three cars in Fig. 6.8. The video sequence of the three persons has 697 frames, and video sequence of the three cars has 237 frames.

6.7. Conclusions and future work

We propose a novel distributed framework for detecting and integrating of inconsistent measurements. The modeling is based on Markov networks. The Bayesian EM inference reveals the iterative integration of the measurements, from which principled criteria are developed to detect inconsistency. We regard measurements which are inconsistent with the majority of their neighbors as false. They will be eliminated and the integration is performed again, i.e., the estimates in those nodes with false measurements will only rely on the measurements from its neighbors. We apply the proposed robust integration framework for part based ensemble tracking and promising results are obtained.

Future work may include the automatic part selection, and better means to handle the integration in unbalanced Markov networks. We are also interested in exploiting the integration framework to other computer vision applications.

CHAPTER 7

Conclusion and future research

The rapid development of image and video sensors (e.g., cheap CCD cameras) and computer hardware (e.g., high speed CPU and huge memory) make it very convenient to acquire, store and process huge amount of visual data, either in the form of images, or in the form of videos. With such huge amount of visual data, it is an emerging need to endow the computers the capability to intelligently perceive and manipulate them automatically. However, although we have observed the boom of the research on computer vision in recent years, it is still in its infancy to build a visually intelligent computer. Among all the challenging computer vision problems, to build a computer to visually analyze complex motion from videos is one of the most basic problems, since it is the fundamental elements for different levels of machine intelligence such as *gesture recognition*, *action recognition* and even for *behavior analysis*. All of these are in turn fundamental elements of many emerging computer vision applications such as intelligent video surveillance, video events analysis, robotics, vision for graphics, etc., to name a few.

Targeting on the fundamental challenges of complex motion analysis, we propose a novel distributed probabilistic approach to complex motion analysis, which clearly demonstrates that a set of interactive motion analyzers working collaboratively to obtain the solution. In answering the first question in Sec. 1.1.1, "*what characterizes the optimal integration of the set of motion analyzers?*". Under our distributed probabilistic representation based on Markov networks, the optimal integration is characterized by either the fixed-point of the

mean field fixed-point equations [**43**, **45**, **126**, **133**] or the convergent results of the belief propagation algorithm [**41**, **47**]. In answering the second question in Sec. 1.1.1, both the mean field variational method and the belief propagation algorithm are iterative local "message" passing algorithms, which are all quite efficient computational diagrams to obtain the optimal integration of the set of motion analyzers. We summarize the work presented in this dissertation followed by the discussions of some of the future work.

7.1. Summary

In summary, we have developed a novel collaborative approach to complex motion analysis from video. The theoretical foundation is based on probabilistic variational analysis on graphical models. Compared to previous work, the new approach is scalable to the increase of the degrees of freedom of complex motions. It reveals a distributed probabilistic information integration framework, which is able to effectively handle *measurement uncertainty*, *measurement multi-modality*, and *measurement inconsistency*, i.e.,

- The measurement uncertainties are captured very well by the probabilistic reasoning on graphical models.
- For measurement multi-modality, we can either keep multiple hypothesis using particle filters when performing the Bayesian inference, or identify the optimal mode of the integrated measurement (e.g., using the proposed VMAP [**43**]).
- For measurement inconsistency, we have proposed rigorous quantitative conditions to decide measurement consistency and inconsistency in the distributed information integration framework based on graphical models. They greatly facilitate to exclude the false measurements from the integration process.

We have also contributed several novel and practical variational Bayesian inference algorithms on graphical models, namely the MFMC algorithm [**45**, **46**, **126**], the VMAP algorithm [**40**], and the DDBP algorithm [**41**], which are able to either obtain the posterior distributions or obtain the MAP estimates on graphical models very efficiently. By incorporating the data driven importance sampling technique into the Monte Carlo implementation of both the mean field fixed-point iterations and the belief propagation updating, we also achieve a principled parallel framework to combine top-down and bottom-up reasoning together. Extensive experimental results on different types of complex motions under various scenarios demonstrate the effectiveness, efficiency and robustness of the proposed collaborative approach.

7.2. Future research

Besides what we have presented in this dissertation, we are quite interested in extending our current work in the following directions:

- As we have mentioned, the methodology of the collaborative approach we proposed is indeed very general and may be applied to solve other computer vision problems besides complex motion analysis. We intend to apply the proposed approach to other emerging applications such as biomedical image analysis [**141**], stereo vision [**112**, **113**], joint audio-visual recognition [**87**], and large-scale sensor networks, with both video and non-video sensors [**62**].
- According to the taxonomy in Chapter 2, we have investigated and demonstrated the powerfulness of exploiting *latent variable inference* on graphical models to solve computer vision problems. Further research is necessary to investigate the capability

of exploiting *parameter learning* and *structure learning* techniques to address some other fundamental issues in computer vision problems.

- The proposed algorithms all exchange "messages" in a local neighborhood. In large scale graphical models, such a local neighborhood message passing strategy may still be slow. We are interested in developing more efficient collaborative and distributed message passing schemes which could speed up the probabilistic inference process. Multi-scale message passing in the model level could be a potential solution, at the highlight of [**26**].
- Exploit the output from the proposed approach to complex motion analysis, we may further develop real systems for other emerging applications such as intelligent video surveillance, perceptual human computer interface, motion capturing for animation, and even for applications based on mobile phone cameras (see an example of business card scanning [**39**]).

We expect future research on these directions to be far-reaching, since we are targeting on either addressing some of the fundamental issues that commonly exist in many computer vision problems, or realizing emerging computer vision applications using the proposed approach.

References

[1] Hirotugu Akaike. A new look at the statistical model identification. *IEEE Transaction on Automatic Control*, 19(6):716–723, December 1974.

[2] Christophe Andrieu and Arnaud Doucet. Joint bayesian model selection and esitimation of noisy sinusoids via reversible jump mcmc. *IEEE Transaction on Signal Processing*, 47(10):2667–2676, 1999.

[3] Adrian Barbu and Song-Chun Zhu. Graph partition by swendsen-wang cut. In *Proc. IEEE International Conference on Computer Vision*, pages 320–329, 2003.

[4] Adrian Barbu and Song-Chun Zhu. Generalizing swendsen-wang to sampling arbitrary posterior probabilities. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 27(8):1239–1253, 2005.

[5] Carlos Barrn and Ioannis Kakadiaris. Estimating anthropometry and pose from a single uncalibrated image. *Computer Vision and Image Understanding*, 81(3):269–284, 3 2001.

[6] Matthew J. Beal. *Variational Algorithms for Approximate Bayesian Inference*. Phd thesis, Gatsby Computational Neuroscience Unit, University College London, 2003.

[7] Matthew J. Beal and Zoubin Ghahramani. The variational bayesian em algorithm for incomplete data: with application to scoring graphical model structures. In *Bayesian Statistics 7*, pages 453–464. Oxford University Press, 2003.

[8] Michael Black and Allan Jepson. Eigentracking: Robust matching and tracking of articulated object using a view-based representation. In *Proc. European Conf. Computer Vision*, volume 1, pages 343–356, Cambridge, UK, 1996.

[9] Andrew Blake and Michael Isard. *Active Contours*. Springer-Verlag, 1998.

[10] Christoph Bregler and Jitendra Malik. Tracking people with twists and exponential map. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 8–15, 1998.

[11] Lars Bretzner, Ivan Laptev, and Tony Lindeberg. Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering. In *Proc. 5th IEEE International Conference on Automatic Face and Gesture Recognition*, pages 423–428, 2002.

[12] Lars Bretzner, Björn Thuresson, and Sören Lenman. Combining hand gestures and flow menus in computer vision based interfaces. In *Proc. of 11th International Conference on Human Computer Interaction*, Las Vegas, NV, July 2005.

[13] Jinxiang Chai and Jessica Hodgins. Performance animation from low-dimensional control signals. *ACM Transactions on Graphics (Proc. of SIGGRAPH 2005),*, 24(3):686 – 696, July 2005.

[14] Tat-Jen Cham and James M. Rehg. A multiple hypothesis approach to figure tracking. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 239–245, Ft. Collins, CO, 6 1999.

[15] Kiam Choo and David Fleet. People tracking using hybrid Monte Carlo filtering. In *Proc. IEEE Int'l Conf. on Computer Vision*, volume II, pages 321–328, Vancouver, Canada, July 2001.

[16] Dorin Comaniciu. Nonparametric information fusion for motion estimation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 59–68, 2003.

[17] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Real-time tracking of non-rigid objects using mean shift. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 142–149, Hilton Head Island, South Carolina, 2000.

[18] T. Cootes, D. Cooper, C. Taylor, and J. Graham. Active shape models - their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[19] James M. Coughlan and Sabino J. Ferreira. Finding deformable shapes using loopy belief propagation. In *Proc. European Conference on Computer Vision*, volume LNCS 2352, pages 453–468, 2002.

[20] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley and Sons, New York, 2 edition, 1991.

[21] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

[22] Jonathan Deutscher, Andrew Blake, and Ian Reid. Articulated body motion capture by annealed particle filtering. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 2126–2133, Hilton Head Island, South Carolina, 6 2000.

[23] Detlev Doll and Werner von Seelen. Object recognition by deterministic annealing. *Image and Vision Computing*, 15:855–860, 1997.

[24] Arnaud Doucet, Nando De Freitas, and Neil Gordon. *Sequential Monte Carlo Methods in Practice*. New York: Springer-Verlag, 1 2001.

[25] Pedro Felzenszwalb and Dan Huttenlocher. Efficient matching of pictorial structures. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 2066–2073, 2000.

[26] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient belief propagation for early vision. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 261–268, 2004.

[27] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 61(1):55–79, January 2005.

[28] W. Freeman, E. Pasztor, and O. Carmichael. Learning low-level vision. *Int'l Journal of Computer Vision*, 40:25–47, 2000.

[29] William T. Freeman and Egon C. Pasztor. Learning low-level vision. In *Proc. IEEE International Conference on Computer Vision*, pages 1182–1189, 1999.

[30] William T. Freeman and Egon C. Pasztor. Markov network for low-level vision. Technical report, MERL, Mitsubish Electric Research Laboratory, 1999.

[31] William T. Freeman and Hao Zhang. Shape-time photography. In *Proc. IEEE conf. on Computer Vision and Pattern Recognition*, volume 2, pages 151–157, 2003.

[32] Brendan J. Frey and Nebojsa Jojic. A comparison of algorithms for inference and learning in probabilistic graphical models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(9):1392–1416, September 2005.

[33] Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 721–741, June 1984.

[34] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter. *Markov Chain Monte Carlo in Practice*. Chapman & Hall/CRC, 1996.

[35] Greg Hager and Peter Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(10):1025–1039, 1998.

[36] Ismail Haritaoglu, David Harwood, and Larry Davis. W4: Who? when? where? what? a real time system for detecting and tracking people. In *Proc. IEEE Int'l Conf. on Face and Gesture Recognition*, pages 222–227, Nara, Japan, April 1998.

[37] Bernd Heisele, Thomas Serre, Massimiliano Pontil, and Tomaso Poggio. Component-based face detection. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 657–662, 2001.

[38] Jennifer A. Hoeting, David Madigan, Adrian E. Raftery, and Chris T. Volinsky. Bayesian model averaging: A tutorial. *Statistical Science*, 14(4):382–417, November 1999.

[39] Gang Hua, Zicheng Liu, Zhengyou Zhang, and Ying Wu. Automatic business card scanning with a camera. In *Proc. IEEE International Conf. on Image Processing*, Atlanta, GA USA, Octobor 2006. To appear.

[40] Gang Hua, Zicheng Liu, Zhengyou Zhang, and Ying Wu. Iterative local-global energy minimization for automatic extraction of object of interest. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 2006. To appear.

[41] Gang Hua and Ying Wu. Multi-scale visual tracking by sequential belief propagation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 826–833, Washington, DC, June 2004.

[42] Gang Hua and Ying Wu. Capturing human body motion from video for perceptual interfaces by sequential variational map. In *Proc. 11th International Conference on Human-Computer Interactiona*, Las Vegas, July 2005. Invited.

[43] Gang Hua and Ying Wu. Variational maximum a posteriori by annealed mean field analysis. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 27(11):1747–1781, November 2005.

[44] Gang Hua and Ying Wu. Measurement integration under inconsistency for robust tracking. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, New York City, NY, June 2006.

[45] Gang Hua and Ying Wu. Sequential mean field variational analysis of structured deformable shapes. *Computer Vision and Image Understanding*, 101(2):87–99, Feburary 2006.

[46] Gang Hua, Ying Wu, and Ting Yu. Analyzing structured deformable shapes via mean field monte carlo. In *Proc. IEEE Asia Conference on Computer Vision*, Jeju Island, Korea, January 2004.

[47] Gang Hua, Ming-Hsuan Yang, and Ying Wu. Learning to estimate human pose with data driven belief propagation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 747–754, 2005.

[48] Giancarlo Iannizzotto. Vision-based human-computer interaction at visilab. In *Proc. of 11th International Conference on Human Computer Interaction*, Las Vegas, NV, July 2005.

[49] Sergey Ioffe and David Forsyth. Finding people by sampling. In *Proc. IEEE International Conference on Computer Vision*, pages 1092–1097, 1999.

[50] Sergey Ioffe and David A. Forsyth. Mixtures of trees for object recognition. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 180–185, 2001.

[51] Michael Isard. PAMPAS: Real-valued graphical models for computer vision. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 613–620, 2003.

[52] Michael Isard and Andrew Blake. Contour tracking by stochastic propagation of conditional density. In *Proc. European Conference on Computer Vision*, volume 1, pages 343–356, 1996.

[53] Michael Isard and Andrew Blake. CONDENSATION - conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.

[54] Tommi S. Jaakkola. Tutorial on variational approximation method. In *Advanced mean field methods: theory and practice*. MIT Press, 2000.

[55] Tommi S. Jaakkola. Tutorial on variational approximation methods. MIT AI Lab TR, 2000.

[56] Omar Javed, Saad Ali, and Mubarak Shah. Online detection and classification of moving objects using progressively improving detectors. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 696–701, San Diego, CA, June 2005.

[57] Omar Javed, Khurram Shafique, and Mubarak Shah. Appearance modeling for tracking in multiple non-overlapping cameras. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 26–33, San Diego, CA, June 2005.

[58] Michael Jordan and Yair Weiss. Graphical models: Probabilistic inference. In *The Handbook of Brain Theory and Neural Network*, pages 243–266. MIT Press, second edition, 2002.

[59] Micheal Jordan, Zoubin Ghahramani, Tommi Jaakkola, and Lawrence Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37:183–233, 2000.

[60] Shanon X. Ju, Michael J. Blacky, and Yaser Yacoobz. Cardboard people: A parameterizedmodel of articulated image motion. In *Proc. of International Conference on Automatic Face and Gesture Recognition*, pages 38–44, Killington, Vermont, 10 1996.

[61] Simon Julier and Jeffrey Uhlmann. A nondivergent estimation algorithm in the presence of unknown correlations. In *Proc. of the American Control Conference*, volume 4, pages 2369–2373, Alberqueque, June 1997.

[62] Sohaib Khan and Mubarak Shah. Consistent labeling of tracked objects in multiple cameras with overlapping fields of view. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 25(10):1355–1360, Octobor 2003.

[63] Zia Khan, Tucker Balch, and Frank Dellaert. An mcmc-based particle filtering for tracking multiple interacting targets. In *Proc. of European Conference on Computer Vision*, volume T. Pajdla and J. Matas of *LNCS 3024*, pages 279 – 290, Prague, Czech Republic, May 2004.

[64] Zia Khan, Tucker Balch, and Frank Dellaert. Mcmc-based particle filtering for tracking a variable number of interacting targets. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 27(11):1805–1918, November 2005.

[65] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi Jr. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[66] Nahum Kiryati, Y. Eldar, and Alfred M.Bruckstein. A probabilistic Hough transform. *Pattern Recognition*, 24(4):303–316, 1991.

[67] Rick Kjeldsen. Exploiting the flexibility of vision-based user interactions. In *Proc. of 11th International Conference on Human Computer Interaction*, Las Vegas, NV, July 2005.

[68] Frank R. Kschischang, Brendan J. Frey, and Hans-Andrea Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transaction on information theory*, 47(2):498–519, 2001.

[69] Mun Wai Lee and Isaac Cohen. Proposal maps driven MCMC for estimating human body pose in static images. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 334–341, 2004.

[70] Stan Z. Li. Robustizing robust m-estimation using deterministic annealing. *Pattern Recognition*, 29(1):159–166, 1996.

[71] Stan Z. Li. *Markov random field modeling in image analysis*. Springer-Verlag New York, Inc., 2 edition, 2001.

[72] X. Rong Li, Yunmin Zhu, Jie Wang, and Chongzhao Han. Optimal linear estimation fusionpart i: Unified fusion rules. *IEEE Transaction on Information Theory*, 49(9):2192–2208, 2003.

[73] Ce Liu, Heung-Yeung Shum, and Changshui Zhang. Hierarchical shape modeling for automatic face localization. In *Proc. European Conference on Computer Vision*, pages 687–703, 2002.

[74] Jun Liu, Rong Chen, and Tanya Logvinenko. A theoretical framework for sequential importance sampling and resampling. In A. Doucet, N. de Freitas, and N. Gordon, editors, *Sequential Monte Carlo in Practice*. Springer-Verlag, New York, 2000.

[75] John MacCormick and Andrew Blake. A probabilistic contour discriminant for object localisation. In *Proc. IEEE International Conference on Computer Vision*, pages 390–395, 1998.

[76] John MacCormick and Andrew Blake. A probabilistic exclusion principle for tracking multiple objects. In *Proc. IEEE International Conf. on Computer Vision*, pages 572–578, Greece, 1999.

[77] John MacCormick and Michael Isard. Partitioned sampling, articulated objects, and interface-quality hand tracking. In *Proc. of European Conf. on Computer Vision*, volume 2, pages 3–19, 2000.

[78] David J.C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 9 2003.

[79] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equations of state calculations by fast computing machine. *J. Chem. Phys.*, 21:1087–1091, 1953.

[80] Greg Mori, Xiaofeng Ren, Alexei Efros, and Jitendra Malik. Recovering human body configurations: Combining segmentation and recognition. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 326–333, 2004.

[81] Thomas Moselund and Erik Granum. A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding*, 81(3):231–268, 2001.

[82] Kevin Murphy, Yair Weiss, and Michael Jordan. Loopy-belief propagation for approximate inference: An empirical study. In *Proc. Fifteenth Conference on Uncertainty in Artificial Intelligence*, pages 467–475, 1999.

[83] Kevin Patrick Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. Phd thesis, Computer Science Division, University of California, Berkeley, 2002.

[84] Radford M. Neal. Slice sampling. *Annals of Statistics*, 31(3):705–767, 2003.

[85] Vladimir Ivan Pavlovic. *Dynamic Bayesian Networks for Information Fusion with Application to Human-Computer Interfaces*. Phd thesis, Department of Electrical and Computer Engieering, University of Illinois at Urbana-Champaign, 1999.

[86] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988.

[87] Gerasimos Potamianos, Chalapathy Neti, Guillaume Gravier, Ashutosh Garg, and Andrew W. Senior. Recent advances in the automatic recognition of audio-visual speech. *Proceedings of the IEEE*, 91(9):1306– 1326, 9 2003.

[88] Jan Puzicha and Joachim M. Buhmann. Multiscale annealing for grouping and unsupervised texture segmentation. *Computer Vision and Image Understanding (CVIU)*, 76(3):213–230, 1999.

[89] Jan Puzicha, Thomas Hofmann, and Joachim M. Buhmann. Deterministic annealing: Fast physical heuristics for real-time optimization of large systems. In *Proc. of the 15th IMACS World Conference on Scientific Computation, Modelling and Applied Mathematics*, Berlin, 1997.

[90] Wei Qu, Dan Schonfeld, and Magdi Mohamed. Real-time interactively distributed multi-object tracking using a magnetic-inertia potential model. In *Proc. IEEE International Conference on Computer Vision*, volume 1, pages 535–540, Beijing,China, Octobor 2005.

[91] Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proc. of IEEE*, pages 257–286, 1989.

[92] Adrian E. Raftery. Choosing models for cross-classification. *American Sociological Review*, 51:145–146, 1986.

[93] Adrian E. Raftery. A note on bayes factors for log-linear contingency table models with vague prior information. *Journal of the Royal Statistical Society, Series B*, 48:249–250, 1986.

[94] Deva Ramanan and David A. Forsyth. Finding and tracking people from the bottom up. In *Proc. of IEEE Conf. on Computer Vision and Patter Recognition*, volume 2, pages 467–474, Madison, WI, June 2003.

[95] Deva Ramanan, David A. Forsyth, and Andrew Zisserman. Strike a pose: Tracking people by finding stylized poses. In *Proc. of IEEE Conf. on Computer Vision and Patter Recognition*, volume 1, pages 271–278, San Diego, CA, June 2005.

[96] Ajit V. Rao, David J. Miller, Kenneth Rose, and Allen Gersho. A deterministic annealing approach for parsimonious design of piecewise regression models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(2):159–173, Feburary 1999.

[97] James M. Rehg and Takeo Kanade. Model based tracking of self-occluding articulated objects. In *Proc. of International Conference on Computer Vision*, pages 612–617, Cambridge, MA, 6 1995.

[98] J. B. Rosen. The gradient projection method for nonlinear programming. part I. linear constraints. *Journal of the Society for Industrial and Applied Mathematics*, 8(1):181–217, March 1960.

[99] J. B. Rosen. The gradient projection method for nonlinear programming. part II. nonlinear constraints. *Journal of the Society for Industrial and Applied Mathematics*, 9(4):514–532, December 1961.

[100] Stefan Roth, Leonid Sigal, and Michael Black. Gibbs likelihoods for Bayesian tracking. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 886–893, 2004.

[101] Gregory Shakhnarovich, Paul Viola, and Trevor Darrell. Fast pose estimation with parameter-sensitive hashing. In *Proc. IEEE International Conference on Computer Vision*, volume 2, pages 750–757, 2003.

[102] Jianbo Shi and Carlo Tomasi. Good features to track. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 593–600, June 1994.

[103] Hedvig Sidenbladh and Michael Black. Learning the statistics of people in image and video. *International Journal of Computer Vision*, 54(1-3):183–209, 2003.

[104] Hedvig Sidenbladh, Michael J. Black, and David J. Fleet. Stochastic tracking of 3d human figures using 2d image motion. In *Proc. of European Conference on Computer*

Vision, volume 2 of *Lecture Notes in Computer Science*, pages 702–718, Dublin, Ireland, June 2000. Springer Verlag.

[105] Leonid Sigal, Sidharth Bhatia, Stefan Roth, and Michael Black. Tracking loose-limbed people,. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 421–428, 2004.

[106] Leonid Sigal, Michael Isard, Benjamin Sigelman, and Michael Black. Attractive people: Assembling loose-limbed models using non-parametric belief propagation. In *Advances in Neural Information Processing System 16*. MIT Press, 2004.

[107] Cristian Sminchisescu, Atul Kanaujia, Zhiguo Li, and Dimitris Metaxas. Discriminative density propagation for 3d human motion estimation. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 390–397, San Diego, CA, June 2005.

[108] Cristian Sminchisescu and Bill Triggs. Estimating articulated human motion with covariance scaled sampling. *International Journal of Robotics Research*, 22(6):371–393, June 2003.

[109] Yang Song, Xiaolin Feng, and Pietro Perona. Towards detection of human motion. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1810–1817, Hilton Head Island, SC, June 2000.

[110] Erik B. Sudderth, Alexander Ihler, William Freeman, and Alan Willsky. Nonparametric belief propagation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 605–612, 2003.

[111] Erik B. Sudderth, Michael I. Mandel, William T. Freeman, and Alan S. Willsky. Distributed occlusion reasoning for tracking with nonparametric belief propagation. In *Advances in Neural Information Processing System 17*, 2004.

[112] Jian Sun, Heung-Yeung Shum, and Nan-Ning Zheng. Stereo matching using belief propagation. In *Proc. European Conference on Computer Vision*, volume 2, pages 510–524, 2002.

[113] Jian Sun, Heung-Yeung Shum, and Nan-Ning Zheng. Stereo matching using belief propagation. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 25(7):787–800, July 2003.

[114] Rawesak Tanawongsuwan and Aaron Bobick. Gait recognition from time-normalized joint-angle trajectories in the walking plane. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 726–731, 2001.

[115] Michael E. Tipping and Christopher M. Bishop. Probabilistic principle component analysis. *Journal of Royal Statistical Society, Series B*, 61(3):611–622, 1999.

[116] Zhuowen Tu and Song-Chun Zhu. Image segmentation by data-driven Markov chain Monte Carlo. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 24(5):657–673, 2002.

[117] Jeffrey K. Uhlmann. Covariance consistency methods for fault-tolerant distributed data fusion. *Information Fusion, Elsevier Science*, 4(3):201–215, 3 2003.

[118] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 511–518, 2001.

[119] Liang Wang, Huazhong Ning, Tieniu Tan, and Weiming Hu. Fusion of static and dynamic body biometrics for gait recognition. In *Proc. of IEEE International Conference on Computer Vision*, pages 1449–1454, 2003.

[120] Yang Wang, Tele Tan, and Kia-Fock Loe. Video segmentation based on graphical models. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 335–342, 2003.

[121] David L. Weakliem. A critique of the bayesian information criterion for model selection. *Sociological Methods and Research*, 27(3):359–397, Feburary 1999.

[122] Andrew Wilson and Nuria Oliver. Multimodal sensing for explicit and implicit interaction. In *Proc. of 11th International Conference on Human Computer Interaction*, Las Vegas, NV, July 2005.

[123] John M. Winn. *Variational Message Passing and Its Application*. Phd thesis, Department of Physics, University of Cambridge, 2003.

[124] C. Wren, A. Azarbayejani, T. Darrel, and A. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 9:780–785, July 1997.

[125] Ying Wu, Gang Hua, and Ting Yu. Switching observation models for contour tracking in clutter. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 295–302, 2003.

[126] Ying Wu, Gang Hua, and Ting Yu. Tracking articulated body by dynamic markov network. In *Proc. IEEE International Conference on Computer Vision*, pages 1094–1101, Nice,Côte d'Azur,France, October 2003.

[127] Ying Wu and Thomas S. Huang. Vision-based gesture recognition: A review. In A. Braffort, R. Gherbi, S. Gibet, J. Richardson, and D. Teil, editors, *Gesture-Based Communication in Human-Computer Interaction*, Lecture Notes in Artificial Intelligence 1739, pages 93–104. Springer-Verlag, 1999.

[128] Ying Wu and Thomas S. Huang. Self-Supervised learning for visual tracking and recognition of human hand. In *Proc. AAAI National Conf. on Artificial Intelligence*, pages 243–248, 2000.

[129] Ying Wu and Thomas S. Huang. View–independent recognition of hand postures. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume II, pages 88–94, 2000.

[130] Ying Wu and Thomas S. Huang. Robust visual tracking by co-inference learning. In *Proc. IEEE Int'l Conference on Computer Vision*, volume II, pages 26–33, 2001.

[131] Ying Wu, John Lin, and Thomas S. Huang. Capturing natural hand articulation. In *Proc. IEEE Int'l Conference on Computer Vision*, volume II, pages 426–432, 2001.

[132] Ying Wu, Ting Yu, and Gang Hua. Tracking appearances with occlusions. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 789–795, Madison, Wisconsin, June 2003.

[133] Ying Wu, Ting Yu, and Gang Hua. A statistical field model for pedestrian detection. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 1023–1030, San Diego, June 2005.

[134] Jonathan Yedidia, William Freeman, and Yair Weiss. Understanding belief propagation and its generalization. In *Exploring Artificial Intelligence in the New Millenium*, chapter 8, pages 239–286. Elsevier Science and Technology Books, 2003.

[135] Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. Generalized belief propagation. In *Advances in Neural Information Processing Systems (NIPS)*, volume 13, pages 689–695. MIT Press, 2000.

[136] Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Trans on Information Theory*, 51(7):2282–2312, July 2005.

[137] Ting Yu and Ying Wu. Collaborative tracking of multiple targets. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 834–841, Washington, DC, June 2004.

[138] A. L. Yuille and J. J. Kosowsky. Statistical physics algorithms that converge. *Neural Computation*, 6(3):341–356, June 1994.

[139] Tao Zhao and Ram Nevatia. Bayesian human segmentation in crowded situations. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 459–466, Madison, Wisconsin, June 2003.

[140] Tao Zhao and Ram Nevatia. Tracking multiple humans in crowded environment. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 406–413, Washington, DC, June 2004.

[141] Xiang Sean Zhou, Dorin Comaniciu, and Alok Gupta. An information fusion framework for robust shape tracking. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 27(1):115–129, 2005.

APPENDIX A

Lemmas of Theorem 4.3.1

Define, for $\sigma > 0$, the quantity

$$\varphi_\sigma(\tilde{\mu}) = E_q\{\log p(\tilde{\mu} + \sigma\mathbf{x})\} = \int_{\mathbf{x}} q(\mathbf{x}) \log p(\tilde{\mu} + \sigma\mathbf{x}) d\mathbf{x}. \tag{A.1}$$

Note that Eq. 4.2 ensures that $\varphi_\sigma(\tilde{\mu})$ is finite provided that σ is small enough, as we will show in the proof of Lemma 2. We propose the following lemmas for Theorem 4.3.1.

Lemma 1. *Under the same conditions of Theorem 4.3.1, we have*

$$KL(q_\sigma^{\tilde{\mu}}(\mathbf{x}) \| p(\mathbf{x})) = \mathcal{C}_\sigma - \varphi_\sigma(\tilde{\mu}), \tag{A.2}$$

where $\mathcal{C}_\sigma$ is a constant relied only on σ.

Proof.

$$\begin{aligned}
KL(q_\sigma^{\tilde{\mu}}(\mathbf{x}) \| p(\mathbf{x})) &= \int_{\mathbf{x}} q_\sigma^{\tilde{\mu}}(\mathbf{x}) \log \frac{q_\sigma^{\tilde{\mu}}(\mathbf{x})}{p(\mathbf{x})} d\mathbf{x} && \text{(A.3)} \\
&= \int_{\mathbf{x}} q_\sigma^{\tilde{\mu}}(\mathbf{x}) \log q_\sigma^{\tilde{\mu}}(\mathbf{x}) d\mathbf{x} - \int_{\mathbf{x}} q_\sigma^{\tilde{\mu}}(\mathbf{x}) \log p(\mathbf{x}) d\mathbf{x} && \text{(A.4)} \\
&= -\log\{(2\pi e)^n \sigma^n\} - \int_{\mathbf{x}} q_\sigma^{\tilde{\mu}}(\mathbf{x}) \log p(\mathbf{x}) d\mathbf{x} && \text{(A.5)} \\
&= \mathcal{C}_\sigma - \int_{\mathbf{x}} q_\sigma^{\tilde{\mu}}(\mathbf{x}) \log p(\mathbf{x}) d\mathbf{x} && \text{(A.6)} \\
&= \mathcal{C}_\sigma - \int_{\mathbf{x}} \sigma^n q_\sigma^{\tilde{\mu}}(\tilde{\mu} + \sigma\mathbf{x}) \log p(\tilde{\mu} + \sigma\mathbf{x}) d\mathbf{x} && \text{(A.7)}
\end{aligned}$$

$$= \mathcal{C}_\sigma - \int_{\mathbf{x}} q(\mathbf{x}) \log p(\tilde{\mu} + \sigma\mathbf{x}) d\mathbf{x} \tag{A.8}$$

$$= \mathcal{C}_\sigma - E_q\{log(p(\tilde{\mu} + \sigma\mathbf{x}))\} \tag{A.9}$$

$$= \mathcal{C}_\sigma - \varphi_\sigma(\tilde{\mu}) \tag{A.10}$$

□ □

Lemma 2. *Under the same conditions of Theorem 4.3.1, we have that for any* $\mu \in \mathcal{R}^n$,

$$\lim_{\sigma \to 0} \varphi_\sigma(\tilde{\mu}) = \log p(\tilde{\mu}), \tag{A.11}$$

Proof. Eq. 4.2 guarantees that, for σ sufficiently small, $\varphi_\sigma(\tilde{\mu})$ is finite for all $\tilde{\mu} \in \mathcal{R}^n$, i.e., if $\sigma < \frac{\sqrt{2}}{2}$, note by parallelogram law $-(\mathbf{x} - \tilde{\mu})^T(\mathbf{x} - \tilde{\mu}) \leq \tilde{\mu}^T\tilde{\mu} - \frac{\mathbf{x}^T\mathbf{x}}{2}$, we have

$$|\varphi_\sigma(\tilde{\mu})| = \left| \int_{\mathbf{x}} q(\mathbf{x}) \log p(\tilde{\mu} + \sigma\mathbf{x}) d\mathbf{x} \right| \tag{A.12}$$

$$= \left| \int_{\mathbf{x}} \frac{1}{(2\pi)^{\frac{n}{2}}} \exp\left(-\frac{\mathbf{x}^T\mathbf{x}}{2}\right) \log p(\tilde{\mu} + \sigma\mathbf{x}) d\mathbf{x} \right| \tag{A.13}$$

$$\leq \int_{\mathbf{x}} \frac{1}{(2\pi)^{\frac{n}{2}}} \exp\left(-\frac{\mathbf{x}^T\mathbf{x}}{2}\right) |\log p(\tilde{\mu} + \sigma\mathbf{x})| \, d\mathbf{x} \tag{A.14}$$

$$= \int_{\mathbf{x}} \frac{1}{(2\pi)^{\frac{n}{2}}\sigma^n} \exp\left(-\frac{(\mathbf{x} - \tilde{\mu})^T(\mathbf{x} - \tilde{\mu})}{2\sigma^2}\right) |\log p(\mathbf{x})| \, d\mathbf{x} \tag{A.15}$$

$$\leq \int_{\mathbf{x}} \frac{1}{(2\pi)^{\frac{n}{2}}\sigma^n} \exp\left(-(\mathbf{x} - \tilde{\mu})^T(\mathbf{x} - \tilde{\mu})\right) |\log p(\mathbf{x})| \, d\mathbf{x} \tag{A.16}$$

$$\leq \int_{\mathbf{x}} \frac{1}{(2\pi)^{\frac{n}{2}}\sigma^n} \exp\left(\tilde{\mu}^T\tilde{\mu} - \frac{\mathbf{x}^T\mathbf{x}}{2}\right) |\log p(\mathbf{x})| \, d\mathbf{x} \tag{A.17}$$

$$= \frac{1}{(2\pi)^{\frac{n}{2}}\sigma^n} \exp\left(\tilde{\mu}^T\tilde{\mu}\right) \int_{\mathbf{x}} \exp\left(-\frac{\mathbf{x}^T\mathbf{x}}{2}\right) |\log p(\mathbf{x})| \, d\mathbf{x} \tag{A.18}$$

$$< +\infty \tag{A.19}$$

By the continuity of $p(\mathbf{x})$, for any $\epsilon > 0$, there exists $\delta = \delta(\tilde{\mu}) > 0$ such that $|\mathbf{x} - \tilde{\mu}| < \delta$ implies $|\log p(\mathbf{x}) - \log p(\tilde{\mu})| < \epsilon$, we have

$$
\begin{aligned}
&|\varphi_\sigma(\tilde{\mu}) - \log p(\tilde{\mu})| \\
&= \left| \int_{\mathbf{x}} q(\mathbf{x}) \log p(\tilde{\mu} + \sigma\mathbf{x}) d\mathbf{x} - \log p(\tilde{\mu}) \right| && \text{(A.20)} \\
&= \left| \int_{\mathbf{x}} q(\mathbf{x}) \left(\log p(\tilde{\mu} + \sigma\mathbf{x}) - \log p(\tilde{\mu}) \right) d\mathbf{x} \right| && \text{(A.21)} \\
&= \left| \int_{|\mathbf{x}| < \frac{\delta}{\sigma}} q(\mathbf{x}) \left(\log p(\tilde{\mu} + \sigma\mathbf{x}) - \log p(\tilde{\mu}) \right) d\mathbf{x} \right. && \text{(A.22)} \\
&\quad \left. + \int_{|\mathbf{x}| > \frac{\delta}{\sigma}} q(\mathbf{x}) \left(\log p(\tilde{\mu} + \sigma\mathbf{x}) - \log p(\tilde{\mu}) \right) d\mathbf{x} \right| \\
&\leq \left| \epsilon \int_{|\mathbf{x}| < \frac{\delta}{\sigma}} q(\mathbf{x}) d\mathbf{x} \right| + \left| \int_{|\mathbf{x}| > \frac{\delta}{\sigma}} q(\mathbf{x}) \left(\log p(\tilde{\mu} + \sigma\mathbf{x}) - \log p(\tilde{\mu}) \right) d\mathbf{x} \right| && \text{(A.23)} \\
&= \epsilon P \left(|\mathbf{x}_q| < \frac{\delta}{\sigma} \right) + \left| \int_{|\mathbf{x}| > \frac{\delta}{\sigma}} q(\mathbf{x}) \left(\log p(\tilde{\mu} + \sigma\mathbf{x}) - \log p(\tilde{\mu}) \right) d\mathbf{x} \right| && \text{(A.24)} \\
&< \epsilon + \int_{|\mathbf{x}| > \delta} \sigma^{-n} q\left(\frac{\mathbf{x}}{\sigma} \right) \left| \log p(\tilde{\mu} + \mathbf{x}) - \log p(\tilde{\mu}) \right| d\mathbf{x} && \text{(A.25)}
\end{aligned}
$$

For the second term in Eq. A.25, we have shown from Eq. A.12 to Eq. A.19 that it is integrable. Moreover, for σ small enough and for any fixed $|\mathbf{x}| > \delta$, it is easy to show that

$$\lim_{\sigma \to 0} \sigma^{-n} q\left(\frac{\mathbf{x}}{\sigma} \right) = 0. \tag{A.26}$$

Therefore, for any fixed $|\mathbf{x}| > \delta$ and for σ small enough, we have

$$\sigma^{-n} q\left(\frac{\mathbf{x}}{\sigma} \right) < \exp\left(-\frac{\mathbf{x}^T \mathbf{x}}{2} \right), \tag{A.27}$$

thus it follows that

$$\sigma^{-n} q\left(\frac{\mathbf{x}}{\sigma}\right) \left|\log p(\tilde{\mu}+\mathbf{x}) - \log p(\tilde{\mu})\right| < \exp\left(-\frac{\mathbf{x}^T\mathbf{x}}{2}\right) \left|\log p(\tilde{\mu}+\mathbf{x}) - \log p(\tilde{\mu})\right|. \tag{A.28}$$

From the integrability condition in Eq. 4.2 and the properness of $p(\mathbf{x})$, it is easy to figure out that the right hand side of Eq. A.28 is also integrable. With Eq. A.26, applying Lebesgue's dominated convergence theorem, we have that the second term in Eq. A.25 goes to zero as $\sigma \to 0$, i.e., for the given $\epsilon > 0$, there exists a $\sigma_1 = \sigma(\epsilon) > 0$ such that when $\sigma < \sigma_1$,

$$\left|\varphi_\sigma(\tilde{\mu}) - \log p(\tilde{\mu})\right| < \epsilon + \epsilon = 2\epsilon. \tag{A.29}$$

Then we have

$$\lim_{\sigma\to 0} \left|\varphi_\sigma(\tilde{\mu}) - \log p(\tilde{\mu})\right| = 0. \tag{A.30}$$

Therefore, Eq. A.11 holds. □ □

Lemma 3. *Under the same condition of Theorem 4.3.1, if a sequence* $\{\tilde{\mu}_\sigma\}$ *is such that*

$$\lim_{\sigma\to 0} \varphi_\sigma(\tilde{\mu}_\sigma) = \sup_{\mathbf{x}} \log p(\mathbf{x}), \tag{A.31}$$

then

$$\lim_{\sigma\to 0} \tilde{\mu}_\sigma = \mathbf{x}^* \tag{A.32}$$

Proof. We need to prove that if $\varphi_\sigma(\tilde{\mu}_\sigma) \to \log p(\mathbf{x}^*)$ as $\sigma \to 0$, it is impossible that there exists $\delta > 0$ such that infinitely often $|\tilde{\mu}_\sigma - \mathbf{x}^*| \geq 2\delta$. Let us firstly assume that such a δ exists, then according to the continuity of $\log p(\mathbf{x})$, there exists a $\epsilon > 0$ such that

$\log p(\mathbf{x}) < \log p(\mathbf{x}^*) - \epsilon$ for $|\mathbf{x} - \mathbf{x}^*| \geq \delta$. For σ small enough, e.g., $\sigma < \frac{1}{3}\delta$, we then have

$$\begin{aligned}
\varphi_\sigma(\tilde{\mu}_\sigma) &= \int_{\mathbf{x}} q(\mathbf{x}) \log p(\tilde{\mu}_\sigma + \sigma\mathbf{x}) d\mathbf{x} && \text{(A.33)} \\
&= \int_{|\mathbf{x}|<\frac{\delta}{\sigma}} q(\mathbf{x}) \log p(\tilde{\mu}_\sigma + \sigma\mathbf{x}) d\mathbf{x} + \int_{|\mathbf{x}|>\frac{\delta}{\sigma}} q(\mathbf{x}) \log p(\tilde{\mu}_\sigma + \sigma\mathbf{x}) d\mathbf{x} && \text{(A.34)} \\
&\leq (\log p(\mathbf{x}^*) - \epsilon)\, P\left(|\mathbf{x}_q| < \frac{\delta}{\sigma}\right) + \log p(\mathbf{x}^*) P\left(|\mathbf{x}_q| \geq \frac{\delta}{\sigma}\right) && \text{(A.35)} \\
&= \log p(\mathbf{x}^*) - \epsilon P\left(|\mathbf{x}_q| < \frac{\delta}{\sigma}\right) && \text{(A.36)} \\
&\leq \log p(\mathbf{x}^*) - \frac{1}{2}\epsilon && \text{(A.37)}
\end{aligned}$$

$$\Longrightarrow$$

$$|\varphi_\sigma(\tilde{\mu}_\sigma) - \log p(\mathbf{x}^*)| \geq \frac{1}{2}\epsilon, \qquad \text{(A.38)}$$

where the Eq. A.35 holds because for $|\mathbf{x}| < \frac{\delta}{\sigma}$, we have $|\tilde{\mu}_\sigma + \sigma\mathbf{x} - \mathbf{x}^*| \geq \delta$ which implies that $\log p(\tilde{\mu}_\sigma + \sigma\mathbf{x}) < \log p(\mathbf{x}^*) - \epsilon$.

Eq. A.38 immediately contradicts with Eq. A.31. Therefore, Eq. A.32 holds given that Eq. A.31 holds. □ □

APPENDIX B

Proof of Theorem 4.3.1

Proof. We firstly proceed to prove

$$\lim_{\sigma \to 0} \sup_{\tilde{\mu}} \varphi_\sigma(\tilde{\mu}) = \sup_{\mathbf{x}} \log p(\mathbf{x}). \tag{B.1}$$

Note that from Lemma 1, the series $\{\tilde{\mu}_\sigma\}$ in Theorem 4.3.1 is also such that

$$\varphi_\sigma(\tilde{\mu}_\sigma) = \sup_{\tilde{\mu}} \varphi_\sigma(\tilde{\mu}) \tag{B.2}$$

First of all, we have $\log p(\mathbf{x}) < \log p(\mathbf{x}^*) + \epsilon$ for any $\epsilon > 0$. Thus for any $\sigma > 0$, we have

$$\begin{aligned} \varphi_\sigma(\tilde{\mu}_\sigma) &= \int_{\mathbf{x}} q(\mathbf{x}) \log p(\tilde{\mu}_\sigma + \sigma \mathbf{x}) d\mathbf{x} && \text{(B.3)} \\ &< (\log p(\mathbf{x}^*) + \epsilon) \int_{\mathbf{x}} q(\mathbf{x}) d\mathbf{x} && \text{(B.4)} \\ &= \log p(\mathbf{x}^*) + \epsilon. && \text{(B.5)} \end{aligned}$$

Moreover, from Lemma 2, we have, for any $\epsilon > 0$, there exists a $\sigma_1 > 0$, for $\sigma < \sigma_1$, we have

$$|\varphi_\sigma(\mathbf{x}^*) - \log p(\mathbf{x}^*)| < \epsilon \tag{B.6}$$

Then, from Eq. B.5 and Eq. B.6, we easily obtain that for $\sigma < \sigma_1$,

$$\log p(\mathbf{x}^*) - \epsilon < \varphi_\sigma(\mathbf{x}^*) < \varphi_\sigma(\tilde{\mu}_\sigma) < \log p(\mathbf{x}^*) + \epsilon. \tag{B.7}$$

Thus, for any $\epsilon > 0$, there exists a $\sigma_1 > 0$, for $\sigma < \sigma_1$, we have

$$|\varphi_\sigma(\tilde{\mu}_\sigma) - \log p(\mathbf{x}^*)| = \left| \sup_{\tilde{\mu}} \varphi_\sigma(\tilde{\mu}) - \sup_{\mathbf{x}} \log p(\mathbf{x}) \right| < \epsilon \tag{B.8}$$

This immediately proves Eq. B.1. Then we can directly conclude that Eq. 4.3 holds by applying Lemma 3. □ □

APPENDIX C

Proof of Theorem 6.3.3

Proof. Fixing σ_{12}^2, Eq. 6.6 guarantees to iteratively obtain the exact MAP estimate on the joint posterior Gaussian. We denote $\hat{\mathbf{x}}_2 = \mathbf{A}_{12}\mathbf{x}_2 + \mu_{12}$ and $\mathbf{S} = \mathbf{P} + \sigma_{12}^2\mathbf{I}$. The convergent results in the E-Step in Eq. 6.6 is the same as,

$$\begin{bmatrix} \mathbf{x}_1 \\ \hat{\mathbf{x}}_2 \end{bmatrix} = \begin{bmatrix} (\sigma_{12}^2\mathbf{I} + \hat{\mathbf{\Sigma}}_2)\mathbf{S}^{-1}\mathbf{z}_1 + \mathbf{\Sigma}_1\mathbf{S}^{-1}\hat{\mathbf{z}}_2 \\ \hat{\mathbf{\Sigma}}_2\mathbf{S}^{-1}\mathbf{z}_1 + (\sigma_{12}^2\mathbf{I} + \mathbf{\Sigma}_1)\mathbf{S}^{-1}\hat{\mathbf{z}}_2 \end{bmatrix}. \tag{C.1}$$

Embedding it to the M-Step in Eq. 6.7, we have

$$\sigma_{12}^2 = \frac{1}{n}\sigma_{12}^2\sigma_{12}^2(\mathbf{z}_1 - \hat{\mathbf{z}}_2)^T\mathbf{S}^{-1}\mathbf{S}^{-1}(\mathbf{z}_1 - \hat{\mathbf{z}}_2). \tag{C.2}$$

Since zero is a solution of σ_{12}^2 for Eq. C.2, we only need to analyze the existence of non-zero solutions of σ_{12}^2 for

$$\frac{1}{n}\sigma_{12}^2(\mathbf{z}_1 - \hat{\mathbf{z}}_2)^T\mathbf{S}^{-1}\mathbf{S}^{-1}(\mathbf{z}_1 - \hat{\mathbf{z}}_2) - 1 = 0. \tag{C.3}$$

Since $\mathbf{P}$ is *real positive definite*, there exists an orthonormal matrix $\mathbf{Q}$ such that $\mathbf{P} = \mathbf{Q}\mathbf{D}_p\mathbf{Q}^T$ where $\mathbf{D}_p = diag[\sigma_1^2, \ldots, \sigma_n^2]$ and $\sigma_1^2 \geq \ldots \geq \sigma_n^2 > 0$. Let $C_p = \frac{\sigma_1^2}{\sigma_n^2}$. We then have $\mathbf{S} = \mathbf{Q}\mathbf{D}_s\mathbf{Q}^T$ and $\mathbf{S}^{-1} = \mathbf{Q}^T\mathbf{D}_s^{-1}\mathbf{Q}$, where $\mathbf{D}_s = diag[\sigma_1^2 + \sigma_{12}^2, \ldots, \sigma_n^2 + \sigma_{12}^2]$ and $\mathbf{D}_s^{-1} = diag[\frac{1}{\sigma_1^2+\sigma_{12}^2}, \ldots, \frac{1}{\sigma_n^2+\sigma_{12}^2}]$. Denote $\tilde{\mathbf{z}} = \mathbf{Q}(\mathbf{z}_1 - \hat{\mathbf{z}}_2) = [\tilde{z}_1, \ldots, \tilde{z}_n]^T$, we have

$$\frac{1}{n}\sigma_{12}^2(\mathbf{z}_1 - \hat{\mathbf{z}}_2)^T\mathbf{S}^{-2}(\mathbf{z}_1 - \hat{\mathbf{z}}_2) = \frac{1}{n}\sum_{i=1}^{n}\frac{\sigma_{12}^2\tilde{z}_i^2}{(\sigma_i^2 + \sigma_{12}^2)^2} \tag{C.4}$$

$$\frac{1}{n}(\mathbf{z}_1 - \hat{\mathbf{z}}_2)^T \mathbf{P}^{-1}(\mathbf{z}_1 - \hat{\mathbf{z}}_2) = \frac{1}{n}\sum_{i=1}^{n} \frac{\tilde{z}_i^2}{\sigma_i^2}. \tag{C.5}$$

From Eq C.4, we only need to analyze the solution of σ_{12}^2 for

$$F(\sigma_{12}^2) = \frac{1}{n}\sum_{i=1}^{n} \frac{\tilde{z}_i^2}{\sigma_i^2} \cdot \frac{1}{2 + \frac{\sigma_i^2}{\sigma_{12}^2} + \frac{\sigma_{12}^2}{\sigma_i^2}} - 1 = 0. \tag{C.6}$$

We proceed to prove the three cases in Theorem 6.3.3.

(a). Eq. 6.8 means $d = \frac{1}{n}\sum_{i=1}^{n} \frac{\tilde{z}_i^2}{\sigma_i^2} > 2 + \sqrt{\frac{\sigma_1^2}{\sigma_n^2}} + \sqrt{\frac{\sigma_n^2}{\sigma_1^2}} \geq 4$. When $\sigma_{12}^2 = k_1 = (d-2)\sigma_1^2$, for any i, we have $\frac{1}{2+\frac{\sigma_i^2}{\sigma_{12}^2}+\frac{\sigma_{12}^2}{\sigma_i^2}} < \frac{1}{2+0+d-2} = \frac{1}{d}$. Thus $F(k_1) < \frac{1}{n}\sum_{i=1}^{n} \frac{\tilde{z}_i^2}{\sigma_i^2} \cdot \frac{1}{d} - 1 = 0$. When $\sigma_{12}^2 = k_2 = \sqrt{\sigma_1^2 \sigma_n^2}$, for any i, $\frac{1}{2+\frac{\sigma_i^2}{\sigma_{12}^2}+\frac{\sigma_{12}^2}{\sigma_i^2}} \geq \frac{1}{2+\frac{\sigma_n^2}{k_2}+\frac{k_2}{\sigma_1^2}} = \frac{1}{2+\sqrt{\frac{\sigma_1^2}{\sigma_n^2}}+\sqrt{\frac{\sigma_n^2}{\sigma_1^2}}} \geq \frac{1}{d}$, thus $F(k_2) \geq \frac{1}{n}\sum_{i=1}^{n} \frac{\tilde{z}_i^2}{\sigma_i^2} \cdot \frac{1}{d} - 1 = 0$. Since $0 < k_2 < k_1$ and $F(\cdot)$ is continuous, there exists a k_3 where $k_2 \leq k_3 < k_1$ and $F(k_3) = 0$. This proves Theorem 6.3.3(a).

(b). Eq. 6.9 means $d = \frac{1}{n}\sum_{i=1}^{n} \frac{\tilde{z}_i^2}{\sigma_i^2} < 4$, then $F(\sigma_{12}^2) \leq \frac{1}{n}\sum_{i=1}^{n} \frac{\tilde{z}_i^2}{\sigma_i^2} \cdot \frac{1}{4} - 1 = \frac{d}{4} - 1 < 0$ for all $\sigma_{12}^2 > 0$. Thus Eq. C.6 has no non-zero solution. Theorem 6.3.3(b) is proven.

(c). Let $F(\sigma_M^2) = \max F(\sigma_{12}^2)$, we show that it must be such that $\sigma_n^2 \leq \sigma_M^2 \leq \sigma_1^2$. Define $F_i(\sigma_{12}^2) = \frac{1}{n}\frac{\tilde{z}_i^2}{\sigma_i^2}\frac{1}{2+\frac{\sigma_i^2}{\sigma_{12}^2}+\frac{\sigma_{12}^2}{\sigma_i^2}}$ thus $F(\sigma_{12}^2) = \sum_i F_i(\sigma_{12}^2) - 1$. Each $F_i(\sigma_{12}^2)$ is monotonically increasing for $0 < \sigma_{12}^2 \leq \sigma_i^2$ and monotonically decreasing for $\sigma_{12}^2 \geq \sigma_i^2$. Therefore $F(\sigma_{12}^2)$ must be monotonically increasing for $0 < \sigma_{12}^2 \leq \sigma_n^2$ and monotonically decreasing for $\sigma_{12}^2 \geq \sigma_1^2$. This tells us that the global maximum of $F(\sigma_{12}^2)$ can only be taken in $\sigma_n^2 \leq \sigma_{12}^2 \leq \sigma_1^2$, thus $\sigma_n^2 \leq \sigma_M^2 \leq \sigma_1^2$. The existence of a non-zero convergent value of σ_{12}^2 implies a non-zero solution for Eq. C.6. We have $F(\sigma_M^2) \geq 0$ otherwise $F(\sigma_{12}^2) < 0$ for all σ_{12}^2 and there is no solution for Eq. C.6. Since $F(0) \to -1$ and $F(\sigma_{12}^2)$ is continuous, there must exist a k_4 such that $0 < k_4 \leq \sigma_M^2 \leq \sigma_1^2$ and $F(k_4) = 0$. This immediately proves Theorem 6.3.3(c). □ □

APPENDIX D

Proof of Corollary 6.3.4

Proof. The Bayesian EM constitutes a fixed-point iteration of σ_{12}^2 in Eq. C.2. From Theorem 6.3.3(c), when non-zero fixed-points exist, at least one of them, $\hat{\sigma}_{12}^2$, is such that $0 < \hat{\sigma}_{12}^2 \leq \sigma_{Pmax}^2 < T(\mathbf{P})$. Then, if the fixed-point iteration is initialized at σ_{pmax}^2 or $T(\mathbf{P})$, it can never surpass $\hat{\sigma}_{12}^2$ to converge to zero since they are scalars. This indicates that σ_{pmax}^2 and $T(p)$ are proper initialization for σ_{12}^2. □ □

Vita

Gang Hua is a Ph.D. candidate in the Department of Electrical Engineering and Computer Science at Northwestern University. His current research interests include computer vision, machine learning, image and video processing, multimedia, visual motion analysis and visual content analysis. He was a research assistant of Prof. Ying Wu at the Computer Vision group of Northwestern University since 2002. During the summer 2005 and 2004, he was a research intern with the Speech Technology Group, Microsoft Research, Redmond, WA, and a research intern with the Honda Research Institute, Mountain View, CA, respectively. Before attending in Northwestern, he was a research assistant in the Institute of Artificial Intelligence and Robotics at Xian Jiaotong University (XJTU), Xian, China. He received his M.S. in pattern recognition and intelligence system at XJTU in 2002. He was enrolled in the Special Class for the Gifted Young of XJTU in 1994 and received his B.S. in Automatic Control Engineering in 1999.

He received the Richter Fellowship and the Walter P. Murphy Fellowship at Northwestern University in 2005 and 2002, respectively. When he was in XJTU, he was awarded the Guanghua Fellowship, the Eastcom Fellowship, the Most Outstanding Student Exemplar Fellowship, the Sea-star Fellowship and the Jiangyue Fellowship in 2001, 2000, 1997, 1997 and 1995 respectively. He was also a recipient of the University Fellowship from 1994 to 2001 at XJTU. He is a student member of IEEE.