Model based tracking of articulated objects

Prabhu, Kaliamoorthi

2014

Prabhu, K. (2014). Model based tracking of articulated objects. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/64795>

<https://doi.org/10.32657/10356/64795>

MODEL BASED TRACKING OF ARTICULATED OBJECTS

PRABHU KALIAMOORTHY

School of Computer Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Doctor of Philosophy

2014

Acknowledgements

I would like to thank my supervisor Assoc. Prof. Ramakrishna Kakarala for his guidance during my PhD. His motivation and support has kept me going through some tough times during my PhD. I believe I have learned lessons from him that would help me through my life. I would like to thank Nanyang Technological University for providing me with the funding and the equipment needed to carryout research.

I would like to express my gratitude to Assoc. Prof. Cham Tat Jen for the stimulating discussions related to my research. I want to thank Assoc. Prof. Quek Hiok Chai for sharing invaluable life lessons to the students of his course work. I would like to express a special thanks to all the universities that provide open course work in the web for helping me get a wide understanding of my research area.

I would like to thank Vittal Premachandran and other research staff in the Game Lab for the interesting arguments and the conversations that made my life pleasant during the period of my PhD.

Finally, I would like to thank my parents, sister, wife, daughter and the rest of my family for encouraging me and being supportive through my PhD years.

Abstract

Human motion capture (HMC) is an active area of research in the computer vision community. The challenging nature of the problem and the diverse application potential has made human pose tracking, which is a fundamental problem for HMC systems, an interesting research area. The approaches to human pose tracking are divided as model free, model based and bottom-up approaches in the current literature. This thesis contributes to model based approach to human pose tracking. Though the state-of-the-art systems in this approach have shown high quality results for human motion reconstruction, they incur a high overhead to achieve this. Hence designing efficient techniques that achieve the same results at an acceptable computational cost has become a crucial research problem, which is addressed in this thesis.

In order to achieve this, three different approaches are pursued in this thesis. We propose a stochastic search method referred to as parametric annealing (PA) for human pose inference. PA improves on the current approaches by reusing function evaluations across annealing layers combined with an appropriate annealing schedule. This results in an efficient inference technique. Furthermore, it relies on a parametric form that enables a more accurate estimation of the latent state.

We also present an improved likelihood formulation for HMC in this thesis. The current methods to HMC use silhouettes and appearance to infer the human pose, which incur a significant overhead due to rendering and matching. We show that by using directional chamfer matching, better tracking results are obtained using significantly lower overhead. Furthermore, we show that the multi-view scenario in which HMC systems operate provide unique opportunities that can be exploited to estimate key parameters of the system.

We further present a technique that enables hierarchical decomposition of the articulated human model. The current systems are not efficient due to the high dimensionality of the state space required to express a human pose. We show that the high dimensional state space can be decomposed into many smaller dimensional subspaces. Furthermore, we describe an efficient

inference scheme that makes use of the decomposed subspaces and achieve state-of-the-art results using a small fraction of the computational resources. The inference scheme we describe is a general optimization procedure that can exploit the structure in the problem and result in efficient optimization.

Contents

Acknowledgements	iii
Abstract	v
1 Introduction	1
1.1 Motivation	3
1.2 Research objective and contribution	4
1.3 Organization of the thesis	6
2 Literature survey	9
2.1 Taxonomy - human pose tracking	9
2.1.1 Model-based approaches	10
2.1.2 Discriminative approach	13
2.1.3 Model free approaches	15
2.1.4 Other approaches	18
2.2 Discussion	19
2.3 Previous work in model based motion capture	19
2.3.1 Local optimization based methods	20
2.3.2 Multiple hypothesis trackers	20
2.3.3 Particle based methods	21
2.3.4 Dimensionality reduction	24
2.3.5 Global optimization techniques from evolutionary computing	25
2.3.6 Other related methods	26
2.4 Summary	26
3 Model-based method for human motion capture	29
3.1 Filtering formulation	30
3.2 Optimization vs filtering	32
3.3 Human shape and pose	32
3.3.1 Shape	33
3.3.2 Pose	33
3.3.3 Kinematic tree	34

3.4	Likelihood	35
3.5	Temporal dynamics	37
3.6	Bayesian Inference	39
3.6.1	Kalman filter	39
3.6.2	Multiple hypothesis trackers	41
3.6.3	Particle filters	43
3.7	Global Optimization	46
3.7.1	Annealed particle filter	46
3.7.1.1	Simulated annealing	47
3.7.1.2	APF overview	48
3.7.1.3	APF formal description	50
3.7.2	Particle Swarm Optimization	52
3.8	Summary	53
4	Stochastic Search	55
4.1	Introduction	55
4.2	Motivation	55
4.3	Parametric Annealing	59
4.3.1	Synthetic example	63
4.3.2	Scaling properties	65
4.3.3	Sensitivity analysis	66
4.4	Tracking framework	68
4.5	Application to human pose tracking	69
4.6	Parameter Selection	73
4.6.1	Overview	73
4.6.2	Random variables and Sampling	75
4.6.3	Optimization	76
4.7	Parameter Selection Results	77
4.7.1	Parameter learning for high tracking accuracy	78
4.7.2	Parameter learning for low tracking overhead	80
4.8	Stochastic search on a manifold	81
4.9	Summary	83
5	A Dynamic Likelihood Model	87
5.1	Introduction	87
5.2	Directional Chamfer Matching	89
5.2.1	Overview	89
5.2.2	Kinematic model and Optimization Procedure	90
5.2.3	Likelihood	91
5.2.4	Complexity and KD trees	92
5.2.5	Dynamic parameter estimation	93
5.3	Experiments and Results	96
5.3.1	Observation Edge Fragments and Model Synthesis	96
5.3.2	Effect of parameter λ	99

5.3.3	Comparison with other cost function	101
5.3.4	Computational Overhead	105
5.4	Summary	106
6	State Space Decomposition	107
6.1	Introduction	107
6.2	State space decomposition	109
6.2.1	Overview	109
6.2.2	Marginal Likelihood	110
6.2.3	Conditional Likelihood	111
6.2.4	Edge fragment prior	114
6.2.5	Inference	115
6.2.6	An illustrative example	116
6.3	Experiments and Results	118
6.4	Summary	124
7	Summary and Future work	125
7.1	Conclusion	125
7.2	Future Work	127
	Appendices	131
A	Appendix A	133
A.1	$\ A - B\ _a$ is a metric	133
A.2	$d_{\mathcal{P}}(A, B)$ is a metric	133
B	Appendix B	135
B.1	Equation 6.2	135
B.2	Equation 6.4	137
B.3	Equation 6.5	137
	Bibliography	139
	List of journal publications	153
	List of conference publications	155

List of Figures

1.1	An active human motion capture system	2
1.2	A marker-less human motion capture system	3
2.1	A typical kinematic model used in top-down approach	11
2.2	Comparison of approximate human model to parametric surface	12
2.3	The typical set of operations performed in discriminative method	14
2.4	A typical loosely assembled human model used in model free methods.	17
2.5	The figure shows the sampling scheme with and without constraining using the IMU sensor data.	23
2.6	The figure shows the latent subspace learned with GPLVM.	25
3.1	A graphical model encoding the assumptions made in hidden markov models	29
3.2	A human limb represented by truncated cone	33
3.3	Likelihood generation explanation	36
3.4	Intuitive demonstration of Kalman filters	41
3.5	Intuitive demonstration of particle filters	44
3.6	A demonstration on the effect of annealing	47
3.7	Intuitive demonstration of APF	49
3.8	Intuitive demonstration of PSO	52
4.1	Test demonstrating effects of reusing samples	56
4.2	An illustrative example demonstrating the effect of annealing schedule	57
4.3	An illustrative example demonstrating the effect on estimate	58
4.4	A comparison of the annealing schedule used by APF and PA	63
4.5	A demonstration of parametric annealing	64
4.6	Scaling properties of pa	65
4.7	Sensitivity analysis for pa	67
4.8	Tracking error for S2 jogging sequence	71
4.9	Tracked model superimposed on the observation	72
4.11	Tracked results obtained using PA ^o	81

4.12	The cost function used for bilateral symmetry plane estimation.	82
4.13	Von mises fisher distribution.	82
4.14	Symmetry plane estimated by stochastic search.	84
5.1	Illustration of directional chamfer matching	88
5.2	Overview of directional chamfer matching	89
5.3	Comparison of OCM and DCM	92
5.4	An illustration of KD trees	93
5.5	Motivation for dynamic directional chamfer matching	94
5.6	Steps involved in extraction of directional edge fragments from the observation	97
5.7	Steps involved in synthesis of directional edge fragments	98
5.8	Dynamically estimated parameters for different camera views of the walking sequence	99
5.9	Approximate camera placements in Human Eva I dataset	100
5.10	Tracking error using dynamic estimation of λ and static λ	100
5.11	Image features used for tracking	102
5.12	Tracking results comparing dynamic directional chamfer matching with alternative likelihoods	103
5.13	Tracking results for the subject 1 jogging sequence obtained using dynamic directional chamfer matching	103
5.14	Qualitative comparison of tracking results obtained using the proposed and the alternative likelihoods	104
5.15	An intuitive reasoning to explain the low computational overhead required for dynamic directional chamfer matching	106
6.1	A motivating example for state space decomposition	108
6.2	A graphical model showing the conditional independence assumptions induced by the kinematic mode	110
6.3	A demonstration of various probabilities used for state space decomposition	112
6.4	A demonstration of camera probabilities for a given part	113
6.5	An illustrative example comparing the stochastic search with and without decomposition	117
6.6	Quantitative tracking results with and without decomposition for various number of samples	117
6.7	Overall tracking results with and without decomposition for various number of samples	119
6.8	Qualitative comparison of tracked output	120
6.9	Tracked results obtained using state space decomposition	123

List of Tables

2.1	Taxonomy of discriminative methods	15
2.2	Taxonomy of model free approaches	18
3.1	Steps undertaken in a Multiple Hypothesis Tracker	42
4.1	Configurations used for validating PA	70
4.2	Overall tracking results for all the configurations	72
4.3	Training and tracking statistics for parameter learning that emphasizes high accuracy	79
4.4	Training and tracking statistics for parameter learning that emphasizes low overhead	80
4.5	Comparative results for stochastic search on a manifold.	84
5.1	Overall tracking error using dynamic estimation of λ and static λ	100
5.2	Overall tracking error for the three likelihoods compared	105
6.1	Overall tracking error for the Human Eva II dataset with and without state space decomposition	122
6.2	Comparison of state space decomposition with prominent methods	122
6.3	Comparison of state space decomposition with related methods	122

LIST OF TABLES

Chapter 1

Introduction

Human motion capture (HMC), which is a technique to capture and reconstruct the motion of a human subject, is an active area of research in the computer vision community. HMC has been studied for more than a decade [1, 2], since numerous applications could benefit from it. In animation, HMC systems [3] can be used to capture realistic motion from humans and quadrupeds, significantly reducing the time duration required for developing animated movies. In surveillance, human tracking [4] can enable activities such as anomalous event detection [5]. In gaming industry, HMC systems [6] can be used as a human computer interaction (HCI) method. HCI methodologies can also be used for controlling virtual and simulation environments [7]. In sports and clinical studies, these methods can be used in motion analysis, for example golf swing analysis [8].

Early research on human motion capture focused on active sensing techniques such as magnetic and electromechanical systems, where the subject is expected to wear a cumbersome and invasive equipment in order to achieve motion sensing. The cumbersome nature of the sensing technique can be observed in Figure 1.1, which shows such a sensing equipment. Though such systems enable motion capture, they have significant drawbacks. They impose physical constraints on the subject, and restrict the type of motion that can be captured. This has resulted in the development of passive methods that are based on computer vision. These optical methods make use of markers to enable motion capture. Passive sensing techniques, despite of being less invasive, still require the subject to wear markers to enable sensing.

CHAPTER 1. INTRODUCTION

Figure 1.1: An active human motion capture system([9]). The figure shows how the equipment can be cumbersome.

Though such systems are suitable for applications such as animation, where the environment and the appearance of the subject can be controlled, many applications such as surveillance cannot rely on markers for motion capture. Furthermore, even if one can attach markers to the subject, a marker-less approach would still be preferred over the marker based one.

The problems in the current systems can be addressed by vision based marker-less motion capture methods that do not expect the subject to wear markers. The advances in the capabilities of the computing systems and the availability of low cost cameras make these methods commercially very attractive. Furthermore, from a purely algorithmic point of view, the problem is considered to be a challenging one. Hence there is an increased research interest in this area among the academic and the research community.

Marker-less motion capture has been researched by the computer vision community for more than a decade [10, 11]. Figure 1.2 shows such a system. It can be observed that the subject does not have markers attached to him. Furthermore, the subject wears regular apparel, i.e., the system does not rely on color coded apparel to enable sensing. Recent studies [12] indicate that marker-less motion capture is within reach. However, the state-of-the-art approach to it that can produce highly accurate motion reconstruction with no prior assumptions, is not close to being a viable alternative to marker

1.1. MOTIVATION

Figure 1.2: A marker-less human motion capture system. The figure shows the ultimate goal of the technique, the subject is not expected to wear special equipment or color coded apparel.

based systems. Furthermore, though application areas such as animation can perform human motion capture as an off-line processing job, a vast majority of the applications such as surveillance and HCI demand real time processing capability. Besides, the advent of Kinect [13], a depth sensor from Microsoft that enables HMC to control avatars in gaming, hints the impending arrival of more consumer grade devices that would rely on HMC in the future. Thus it no longer suffices for these systems to function, but also to function with low computational overhead in consumer grade hardware that are expected to be of low cost. As a result, there is a necessity to further research in human pose tracking which is a fundamental problem in marker-less human motion capture.

1.1 Motivation

A human body is a non-rigid articulated object with many degrees of freedom. Most studies in the current literature model it as a rigid articulated object in order to make it tractable. Furthermore, the problem of human motion reconstruction is formulated as an optimization or a filtering problem. The current studies model the human skeleton with 25 to 50 degrees of freedom in order to represent a realistic human pose in 3D. As a result of the high dimensionality of the human pose, well known optimization and filtering methods are not well suited for human motion capture.

CHAPTER 1. INTRODUCTION

Some studies have proposed novel optimization methods that are suited for high dimensional problems. These particle based methods use simulations to estimate the optimal pose in an analysis by synthesis framework. Recent evaluation of the method has shown exceptional human motion reconstruction results [14, 15] with visual input. However, the simulations which are required for the optimization incur a significant computational overhead. Hence there is a need to improve the optimization and filtering tools that are used with such problems.

Furthermore, the degrees of freedom of a human skeleton is largely parametrized by joint angles. As a result, the state transitions that occur during filtering are nonlinear resulting in a multi-modal posterior even if the observation model is unimodal. As a result, the filtering tools used with human motion capture needs to handle multi-modal distributions. From an optimization perspective, this would require global optimization tools that can handle a non-convex objective.

These aspects make human motion capture a challenging problem from an algorithmic perspective. Before describing the main objective of this research, we want to mention that, though human motion capture is the main application area for this study, this problem falls in the general class with same complexity referred to as articulated body motion capture. In addition, the methods we discuss such as efficient optimization methods for high dimensional non-convex problems are applicable to many other problems apart from articulated body motion capture.

1.2 Research objective and contribution

This study aims to develop efficient methods for human pose estimation and tracking from visual data. Human pose estimation can be defined as the problem of estimating the pose parameters of a human model that provides the best explanation for the pose of a subject in the observed image or a set of images from various viewpoints. Human pose tracking can be defined as the problem of dynamically estimating the human pose as it changes with time, accounting for the ambiguities in the pose caused by uncertainties in the observed image. The study operates under the model-based approach

1.2. RESEARCH OBJECTIVE AND CONTRIBUTION

(described in Section 2.1.1) to human pose tracking within the Bayesian filtering framework (described in Section 3.1).

Recent studies [15] indicate a number of areas within the framework that needs further research. The inference method is identified to be one of the most important among them. As described earlier, the state space required to express a human pose is high dimensional. As a result, existing inference techniques known to the community incur a high overhead to reconstruct the human motion. The study in [14] report that the state of the art system in this area takes up to 76 seconds to process a single frame. This is an unacceptably large overhead for commercial systems. A suitable inference method can significantly reduce the overhead required for these systems. In this thesis, we propose an improved inference technique that performs better than the current methods.

Another important aspect of the human motion capture system is the likelihood model or the cost function. The study in [15] indicates that a suitable likelihood function would improve both the reconstruction accuracy and the computational overhead incurred by these systems. Current systems to HMC [15, 14] use a cost function constructed with silhouettes and an appearance model of the subject. In this thesis, we propose a dynamic likelihood model for human motion capture using directional chamfer matching. We show that it performs significantly better than the cost functions used in recent studies such as [15, 14].

The current methods in the literature formulate the problem of human motion capture as either an optimization or a filtering problem. This is achieved by defining a joint likelihood over the parameters of the kinematic model. Since the dimensionality of the problem is high, a systematic approach that performs a decomposition of the state space would result in an efficient inference scheme. Besides, such a decomposition can also avoid the combinatorial explosion of the number of hypothesis in the joint space. In this thesis, we describe a formulation of articulated body motion capture, that decomposes the states space and enables the inference method to exploit the hierarchical structure in the problem. This results in an efficient inference scheme that has significantly lower overhead than the current methods.

1.3 Organization of the thesis

The rest of the thesis is organized as follows.

Chapter 2 provides a literature survey on human pose tracking and estimation. A taxonomy of various approaches pursued in the area is provided in this chapter in order to outline the approach pursued in this thesis. Each approach including the one pursued in this thesis is described and some prominent studies in them are discussed in detail.

Chapter 3 introduces the model based approach for human pose tracking, which is the approach pursued in this thesis. Various terminologies relevant to the approach are introduced here. The filtering formulation for human pose tracking is derived and its relation to the optimization formulation is elucidated. Various aspects of the framework such as the human pose, shape and likelihood are described along with suitable examples to illustrate them. Various inference methods used for tracking are also described along with illustrative examples. The merits and demerits of each inference technique is discussed.

Chapter 4 describes a novel inference procedure referred to as parametric annealing (PA) [16, 17]. The drawback in the existing methods are outlined to motivate the need for PA. The procedure is described in detail with a contrived example to illustrate it. A quantitative comparison of PA with the current methods is performed using the Human Eva I dataset. Furthermore, an automated framework to enable the parameter selection for various stochastic search procedures are discussed here. An extension of the stochastic search methods to the manifold $\mathbb{S}^n$ [18] is also discussed along with an application to the problem of bilateral symmetry plane estimation for a 3D surface.

Chapter 5 describes a likelihood model for human motion capture using dynamic directional chamfer matching [19, 20]. The chapter also describes how the multi-view nature of human motion capture system can be exploited to estimate key parameters for the likelihood. The proposed dynamic oriented chamfer matching method is validated using the Human Eva I dataset.

Chapter 6 describes a state space decomposition framework [21] for human motion capture. The details of the framework such as the inde-

1.3. ORGANIZATION OF THE THESIS

pendence assumptions that it makes and the conditional likelihoods for the decomposed subspaces are provided with suitable figures that elucidates the formulation. The decomposition framework is validated by comparing it with the current methods using the data from the Human Eva I and II dataset.

Chapter 7 provides the concluding remarks for the thesis. The future research directions that needs to be pursued in order to enable a marker-less HMC system that is commercially viable is also outlined in this chapter.

CHAPTER 1. INTRODUCTION

Chapter 2

Literature survey

In previous chapter, we discussed a number of applications that would potentially benefit from research in human pose tracking. Though it is desirable to have a single approach that would address all these applications, such an approach is not feasible due to the varying requirements imposed by each application. As a result, in the current literature, there are several approaches to human pose tracking. In this chapter, we first provide a brief literature survey on this area. We summarize various research in the area and provide a taxonomy of the approaches in the current literature. Then we briefly discuss various methods and point out the approach pursued in this study, namely model-based approach. This is followed by a discussion on various prominent model-based methods in the current literature. The advantages and the disadvantages of each method is also discussed.

2.1 Taxonomy - human pose tracking

The existing literature on human pose tracking is broadly divided as model-based approaches [22, 23, 24, 25, 26, 27, 28, 29, 30, 31], discriminative approaches [32, 33, 34, 35, 36, 37, 38] and model free [39, 40, 41, 42, 43, 44] approaches. They are described briefly in this section. A detailed survey of these methods are available in [1, 10].

2.1.1 Model-based approaches

In many applications such as animation, sports and medical motion analysis an accurate method to reconstruct the motion of a subject is necessary. Model-based approaches in the current literature addresses this need. Model-based approaches (also referred to as generative method, analysis by synthesis technique, and top-down approach) formulate the problem of human pose tracking as an optimization or a filtering problem. The skeleton of the subject to be tracked is modeled using a kinematic model [45]. A typical kinematic model used in such systems is shown Figure 2.1. Though the kinematic model is custom made in most research, studies such as [46, 47, 48] present an automated method to build the kinematic skeleton. The degrees of freedom of the kinematic model such as the global translation, rotation and joint angles parametrize the model. Most studies use a kinematic model with 25 to 50 degrees of freedom.

The kinematic skeleton is fleshed to create a model that accurately resembles the subject being tracked. Figure 2.2 shows a typical model fleshed with truncated cones superimposed on the subject being tracked. Earlier work on this approach mostly relied on approximate models using truncated cones or ellipsoids. However, modern work in the area [49, 50] often rely on parametric or exact surface meshes to flesh the kinematic skeleton. Figure 2.2 shows the difference between approximate and parametric surface model of the subject. It can be observed from the figure that using surface meshes result in better tracking results.

The approach typically operates in a multi-view scenario and all the cameras are assumed to be calibrated with respect to the world coordinate system. The model is initialized using other approaches such as discriminative approach [51] or model-free approach [52]. The initialization phase estimates both the initial pose and the shape of the subject [51]. After initialization, the problem of pose estimation for a time frame is formulated as an optimization problem that minimizes a measure of disparity between the image observation and a synthesized projection as a function of the parameters of the kinematic model. Formally, this is expressed as

$$\hat{X} = \arg \min_X \text{diff}(\Pi(X), O) \quad (2.1)$$

2.1. TAXONOMY - HUMAN POSE TRACKING

Figure 2.1: A typical kinematic model used in model based approach [26]. The joints are indicated by the dots and the bones are indicated by straight lines. The model has 17 segments and 29 degrees of freedom.

where $X \in \mathbb{R}^d$ is a vector of parameters of the kinematic model henceforth referred to as the pose parameters, $\Pi(X)$ is the synthesized projection, O is the image observation, $diff$ is a measure of the disparity between the two input and $\hat{X}$ is the estimated pose. Solving (2.1) is a challenging problem for several reasons. Function $diff$ is a multi-variate function of dimensionality in the range of 25 and 50. Apart from the 3 parameters for global translation, all the other parameters needed to describe a human pose are angles and they are related to the output non-linearly. As a result, $diff$ is a non-convex function.

Finding the global minimizer of a non-convex high dimensional function is known to be hard. Furthermore, both the synthesis $\Pi(X)$, as well as the disparity measure $diff$ are compute intensive. Hence, every evaluation of the function would add a significant computational overhead. In order to overcome these difficulties, approximate methods such as multiple hypothesis tracking [25, 22], particle filtering [27] and annealed particle filtering [53, 54] have been used to find an approximate solution to equation (2.1). At a fundamental level these approaches aim to avoid an exhaustive search for the optimal solution to Eq. (2.1). This is achieved using the filtering

CHAPTER 2. LITERATURE SURVEY

Figure 2.2: Difference between approximate model and parametric surface mesh [49]. The columns correspond to the silhouettes obtained by background subtraction, approximate model, synthesis of approximate model superimposed on the silhouette, surface model and synthesis of surface model superimposed on the silhouette respectively. It can be observed that a surface model produces a more accurate estimate when compared to an approximate model.

formulation of the problem, which is described in detail in Section 3.

Model-based approach is arguably the most accurate technique to reconstruct the motion of a human subject in the current literature. This is evident from the results reported by recent 3D model free methods [55] in comparison to recent model based methods [14]. Though recent discriminative methods [56] have shown competitive results, this comes at the cost of significant prior knowledge in terms of learning. Despite the reconstruction accuracy, it has a number of drawbacks. The method has a very high computational overhead. Furthermore, it requires calibrated stationary cameras and is currently confined to indoor scenarios.

2.1. TAXONOMY - HUMAN POSE TRACKING

2.1.2 Discriminative approach

In several applications such as surveillance, animation and gaming, there is a need to obtain an approximate estimate of the initial pose. Furthermore, those initial poses typically occupy a small region in the pose space. In such situations, discriminative methods could be used for pose estimation. Discriminative methods, which are also referred to as bottom-up approaches [33], formulate the problem of pose estimation as a regression problem. They use a set of labeled examples and learn a mapping function between a set of features extracted from the image observation and the latent human pose. One can estimate the pose for a novel observation by extracting the same set of image features and estimating the output for them using the mapping function. These methods are closely related to exemplar based approaches [36, 37, 57]. Figure 2.3 shows the typical sequence of operations that happen in a discriminative approach.

Research studies in discriminative approach aims to learn a mapping function from different set of features such as shape context [58], SIFT [59] and Histogram of Oriented Gradients [60, 38]. It commonly uses a pattern classification algorithm such as nearest neighbor [61] or a regression technique such as relevance vector regression [62], Bayesian mixture of experts [63]. Table 2.1 presents the taxonomy of the discriminative methods. Some prominent studies on discriminative approach is discussed next.

Mori and Malik [36] present an exemplar based method that uses nearest neighbor regression [61] to estimate the pose. They use a shape context [58] based image feature to define a similarity measure between a novel silhouette and labeled silhouettes (exemplars). Nearest neighbor regression which finds the K nearest neighbors for the novel input and interpolates their label is used for pose estimation. Shakhnarovich et al. [57] present an exemplar based technique to pose estimation. The image features extracted from the exemplars are generally high dimensional. As a result, finding the nearest neighbor can be an expensive operation. They use locality sensitive hashing (LSH), which is an approximate method to find nearest neighbors, to make the search tractable. LSH uses a number of hashing functions that map neighboring features to the same hashing index with high probability, thereby enabling sub-linear time (of the number of exemplars) search for the

CHAPTER 2. LITERATURE SURVEY

Figure 2.3: A typical set of operation in discriminative method [34]. (a) shows the input image (b) shows a grid(subsampling) where features are extracted. (c) shows the SIFT features extracted from the input image (d) shows the SIFT features after those that look similar to background are suppressed (e) shows the pose estimated by the mapping function for the selected features.

nearest neighbors.

Agarwal and Triggs [33] present an approach to human pose estimation using relevant vector regression [62]. Relevant vector regression is a technique that couples support vector machine (SVM) [64] in a hierarchical Bayesian framework. The author [62] shows that such a formulation results in a sparse set of support vectors, i.e., smaller subset of the training examples are needed for prediction. In [33], Agarwal and Triggs apply relevant vector regression for human pose estimation and show that it indeed requires a small subset of the training examples for pose estimation, unlike exemplar based techniques.

2.1. TAXONOMY - HUMAN POSE TRACKING

Study with year	Features	Inference
[36] 2002	Shape context	Nearest neighbor
[33] 2004	Shape context	Relevance vector regression
[32] 2005	Shape context	Bayesian mixture of experts
[34] 2006	SIFT	Classical single-valued regressor
[51] 2007	Shape context and radial distance function	Mixture of experts
[38] 2010	HoG	Twin Gaussian Process

Table 2.1: Taxonomy of discriminative methods.

Sminchisescu et al. [32] present a density propagation framework for discriminative approach similar to the Bayesian filtering formulation for the top down approach [27]. Their formulation enables the uncertainty in the pose estimate to be propagated in time. Bo and Sminchisescu [38] propose twin Gaussian processes for discriminative human pose estimation. They observe that regressors such as SVM [64] and RVM [62] do not exploit the correlation in the output variables and present a method that exploits it. Shotton et al. [6] present a discriminative approach using regression forests [65] that estimates the human pose from depth data.

Bottom-up approach though not the most accurate, have the advantage of not requiring prior knowledge in the form of initialization and is capable of tracking large changes in pose. However, when the range of human pose is very large, it would require a large number of training examples and the performance of this method with such wide range of human poses is unknown [10]. Furthermore, these methods are expected to be sensitive to the appearance of the subject, depending on the image feature.

2.1.3 Model free approaches

In applications such as surveillance and tracking people in sports videos such as baseball, often an accurate reconstruction of the human motion is unnecessary. Only a crude estimate of the human limbs in 2D would suffice. Furthermore, many applications such as pedestrian tracking would require

CHAPTER 2. LITERATURE SURVEY

multiple subjects to be tracked efficiently in real-time. Model-free methods in the current literature are well suited for these application scenarios.

Model free methods can be dated back to Pictorial Structures [66] of the 70s which model objects by both the part appearances and the geometric interconnections between parts. The method models an object with L parts, each having its own location variable l_i , $i \in \{1, \dots, L\}$. The variable l_i could include parameters such as pixel coordinates, part orientation and scale. An object is located in an image by optimizing a cost function expressed below

$$\mathcal{L}^* = \arg \min_{\mathcal{L}} \sum_{i=1}^L \left(m_i(l_i) + \sum_{j \in \mathcal{N}_i} \psi(l_i, l_j) \right) \quad (2.2)$$

where $\mathcal{L}$ is the set of all the locations l_i , $i \in \{1, \dots, L\}$, $\mathcal{N}_i$ is a set of neighbors of i , $m_i(l_i)$ is a cost function that measures the fit of the location l_i , $\psi(l_i, l_j)$ is a constraint (can be considered to be a spring) that binds the neighboring locations together. If the constraints induced by $\mathcal{N}_i$ form an acyclic tree, the cost function can be efficiently optimized using dynamic programming. Though the original Pictorial Structures [66] use an energy minimization formulation that we just described, it has been extended with a probabilistic formulation [39] that enable learning of part appearances and inference in a single framework. A typical template made of rectangular patches used in the method is shown in Figure 2.4.

A majority of model free approaches estimate human pose in 2D [39, 42, 43]. However, it has been extended to infer loosely assembled human poses in 3D [40, 41, 67]. The exact method used for part detection vary in each study including eigen-features [41] and boosting based detector [68, 69]. Similarly, the inference method used in each study differs including non-parametric belief propagation [70], MAP estimation using dynamic programming [42, 39] and loopy inference [43, 71]. Table 2.2 shows the taxonomy of the model free approaches. Some prominent studies on model free approach is discussed next.

Felzenszwalb and Huttenlocher [39] present a model free approach to articulated pose estimation from a single image. They extend the traditional Pictorial Structures framework with a probabilistic formulation and show that by restricting the class of geometric constraints, near linear complexity

2.1. TAXONOMY - HUMAN POSE TRACKING

Figure 2.4: A typical loosely assembled human model used in model free methods [43]. Each rectangle correspond to a part defined by an appearance template, the approach finds a configuration of parts such that each part matches the appearance of part template and the geometric constraints between parts hold.

for inference in the number of search grid locations can be achieved. Wu and Nevatia [42] present an approach to detect multiple subjects in a single image using a model free approach. They use a boosting based method using oriented edges for part detection.

Ramanan and Forsyth [43] present a model free approach that relies on a strong appearance model. The appearance of each subject is learned and is used to locate the subject in different image frames. Their formulation relies on a cyclic graph in order to resolve ambiguities in appearance across time frames. Hence they rely on a loopy inference procedure. Ramanan et al. [72, 44] further extend their work and propose a learning based approach for part detection with direct application to model free methods.

Sigal et al. [40, 41] present a model free approach that is capable of tracking human pose in 3D. Since the pose parameters in 3D are continuous random variables, they use a particle based approximation of belief propagation [73, 70] for inference. Their approach allows the integration of top-down and bottom-up features. Burenus et al. [67] present a 3D model free method for human pose tracking. Rather than resorting to randomized methods [40, 41], they discretize the 3D pose space and perform inference using a max product algorithm which is fundamentally same as dynamic programming [39].

CHAPTER 2. LITERATURE SURVEY

Study with year	Part Detection	Inference
[75] 2012	Simulated detector and eigen-features	Non-parametric belief propagation
[76] 2007	Custom method	Loopy inference
[77] 2007	Boosting based detector	A custom greedy method
[39] 2005	Gabor filters etc	MAP estimation using dynamic programming

Table 2.2: Taxonomy of model free approaches.

Corazza et al. [55] present a model free approach using an articulated iterated closest point (ICP) algorithm. ICP is a method used for the registration of contour fragments and 3D point clouds. The algorithm alternates between finding the closest points for each fragment and minimizing the distances between the estimated closest points. In [55] authors incorporate soft constraints between interconnected parts of the articulated object, which is minimized in the second step of ICP using Levenberg-Marquardt [74] algorithm.

Though model free approaches can only provide a crude estimate of the human pose, they have significant advantages. In long shots, it has been shown to perform quite well [42]. In crowded scenes these methods have been shown to track multiple humans [42]. As a result of recent advances in face/part detection [69], these algorithms can be quite fast. They take care of occlusion and estimate a pose even when only part of the human body is visible. The parts that are not visible in the observation are hallucinated to be present [44]. They do not make assumptions about static background or even stationary camera, as long as the part detectors can generate weak hypothesis (need not be accurate [40]) they infer a plausible human pose.

2.1.4 Other approaches

Other approaches to human tracking which rely on color, texture, motion segmentation and approximate humans by a set of blobs in 2D have been pursued in last decade [78, 79]. However, this is not relevant to marker-less human motion capture. Hence it is not included in this literature survey. More information about them can be found in [1, 10].

2.2 Discussion

As discussed in previous section, there are three major approaches to human pose tracking. Model free approaches are shown to perform well in the area of pedestrian detection and coarse human pose estimation in 2D. A line of research from model free methods specifically [40, 41] have shown positive results in 3D. However, they are not very accurate since they do not have a kinematic model that supports the hypothesis. A typical human pose estimated by model free approach can suffer from problems such as change in the distance between directly connected limbs. Though this would suffice applications such as surveillance, this is not sufficient for applications such as animation, sports and medical motion analysis, which require a more accurate reconstruction of the human motion.

Model based approaches on the other hand has shown promising results in recent quantitative studies [15, 12] and are known to be the most accurate way of estimating human pose in 3D. It is postulated that generative and discriminative [51] methods working in complementary fashion is a likely solution that would result in completely automated marker-less human motion capture. Hence in this research, we pursue a model-based approach to human motion capture.

2.3 Previous work in model based motion capture

In this section, we present various prominent studies on model based approach to human pose tracking. As mentioned in Section 2.1.1, the main challenge in model based approach is the dynamic optimization of a high dimensional non-convex function. The existing literature has number of ways in which the problem is addressed. A group of research studies rely on local optimization, by using high quality image observation. Some studies aim to analytically approximate the likelihood and track multiple pose hypothesis in a Bayesian framework. A vast majority of the research rely on statistical particle based methods to track or optimize the high dimensional multi-modal likelihood. Apart from these, there are studies that aim to reduce the dimensionality of the state space in order to make it more tractable and other miscellaneous studies that rely on global optimization methods from

CHAPTER 2. LITERATURE SURVEY

evolutionary computing. We have organized these methods into subgroups, which are described in detail here.

2.3.1 Local optimization based methods

Bregler and Malik [29, 80] formulate the problem of pose estimation as that of an optical flow estimation. The change in the image content due to an articulated object is assumed to be caused by underlying changes in the pose parameters of a kinematic model. They use product of exponential maps (twist representation [45]) to model the subject. The imaging process is modeled as a scaled orthographic projection [81]. These two choices enable a closed form solution to be derived for the estimation of the rigid body parameters using a least squares technique. The estimation method for individual rigid body parameters is extended to include kinematic constraints in a multi-view scenario enabling a closed form solution for tracking the complete articulated pose. Though their approach is an elegant solution for articulated object tracking, it uses a local optimization based method which is expected to fail when large change in pose is observed.

Ballan et al. [82] present a local optimization based approach to HMC and for tracking interacting hands [83]. They use silhouette and optical flow to construct an energy function that is minimized by Levenberg-Marquardt [74] algorithm. Though their approach produces good tracking results, they use an exact model and chroma-keying to achieve this. Exact model is not realistic in many situations, a statistical model [84, 85] can generalize better. However, this would result in an inexact model where local optimization is expected to fail [3]. Furthermore, many applications cannot rely on chroma-keying, which would further increase the uncertainty and multimodality in the likelihood.

2.3.2 Multiple hypothesis trackers

Cham and Rehg [25] model the human subject by 2D Scaled Prismatic Model with 38 dimensional state space and use a generative formulation to track. They observe that the pose hypothesis of an articulated object such as a human is multi-modal and hence use a multiple hypothesis tracker.

2.3. PREVIOUS WORK IN MODEL BASED MOTION CAPTURE

The tracker models the posterior as a novel piecewise Gaussian distribution and assume a constant velocity model for temporal propagation. For each new observation, samples are drawn from the prior which is obtained after applying a motion model to posterior from last time instance. Then the samples are locally optimized till convergence to the nearest mode using Gauss Newton method. The converged samples are used to estimate the parametric form of the likelihood as a piecewise mixture of Gaussian which is multiplied by the prior to obtain the posterior distribution parametrically. Their approach requires closed form expressions for gradient computation, which severely restrict the choice of likelihood. More modern methods use sampling based techniques that overcome these problems.

Sminchisescu and Triggs [22] present a multiple hypothesis tracking approach similar to [25]. They observe that the alternative modes are most likely to occur along the direction in which the distribution is most uncertain, hence by sampling from the inflated distribution, the samples are most likely to converge to new modes. They use a second order method similar to newton's method for local optimization, they argue that though these methods require the second derivative, once converged they could be used to estimate the variance of the modes. However, second derivatives would require a closed form expression for the likelihood, which significantly restrict the applicability of these methods.

Zhang and Seah [86] use a multiple hypothesis tracking approach i.e. local optimization and filtering, with roots from evolutionary computing. They use a voxel based likelihood to reconstruct human pose without using a strong motion prior. The method is compared to Annealed Particle Filter (APF) and their implementation is shown to run in real-time at 4 - 9 frames per second. The accuracy of the tracker is shown to be better than APF. Their method uses shape from silhouette, which in general requires many camera views and reliable silhouettes for good reconstruction accuracy.

2.3.3 Particle based methods

Sidenbladh, Black and Fleet [27] present a model based approach to pose tracking. Instead of representing the posterior parametrically, they use a non-parametric method, specifically particle filters [87], to track the human

CHAPTER 2. LITERATURE SURVEY

pose. They experiment with an action specific and a constant velocity temporal propagation model. They model the appearance with templates rather than resorting to silhouettes or edges. Though particle filters are capable of tracking an arbitrarily complex distribution, for tracking human pose they observe that approximately 10,000 particles are needed. Since each particle correspond to an evaluation of the image likelihood and these evaluations incur a significant overhead, the overhead to human pose tracking can be measured using the number of particles. The overhead to tracking is high in [27] since it requires 10,000 particles.

Deutscher, Blake, and Reid [53] present a model based approach similar to [27]. However, they extend particle filters with annealing to track the human pose. They use a sparsely sampled silhouette and edge map of the image observation from 3 cameras to define a minimization objective similar to equation (2.1). They show successful tracking results with as few as 2000 particles. This will be discussed further in Chapter 3.

Choo, Poon and Fleet [88, 89] present an approach to human pose tracking using hybrid Monte Carlo filter. Hybrid Monte Carlo [90] filters use Hamiltonian dynamics to construct a Markov transition kernel that leaves any distribution invariant, provided the respective density and the gradient of the density can be evaluated at any point in the state space. When the distribution is considered to be the image likelihood, a set of samples are efficiently generated that approximates the image likelihood. In [88], the authors show that using hybrid Monte Carlo filter, their system is able to track using a very small fraction of the samples in comparison to particle filters. However, hybrid Monte Carlo filter requires analytical gradient, which restricts the image likelihood model that can be used with their method.

Xu and Li [28] infer only a part of the human pose (right half and global six degrees of freedom) using a top-down method while the other half is predicted from the first half using regression, thus exploiting the strong correlation between the right and the left half of the human body during articulated motion such as jogging and walking. They use Rao-Blackwellized particle filter (RBPF [91]), that decomposes the state space into two set of variables. Namely, the root variable and the dependent variable. The decomposition is such that the distribution of dependent variable conditioned on the root variable is analytically tractable. Using the decomposed state

2.3. PREVIOUS WORK IN MODEL BASED MOTION CAPTURE

Figure 2.5: The figure [95] shows the sampling scheme with and without constraining using the IMU sensor data. The sub-figure (a) shows the image observation, (b) shows the samples generated during global optimization, and (c) shows the samples generated that is constrained by the IMU sensor input.

space, RBPF uses a sequential Monte Carlo tracker for the root variable and an alternate tracker such as Kalman filter or Hidden Markov Model [91] for the dependent variable. In [28], authors consider the right half and the global degrees of freedom of the human subject to be the root variable, which is tracked with particle filter. The left half is considered to be the dependent variable, which is tracked analytically using a bank of Kalman filters for each sample of the root variable. Their method cannot provide a general solution to human motion capture, since they rely on action specific information.

Gall, Rosenhahn, Brox, and Seidel [14] demonstrate a human motion capture system using global optimization, filtering and local optimization. They use an improved global optimization technique [92] that is derived from [26]. Filtering is performed by an auto-regressive model using Gaussian process regression [93]. Local optimization is performed using a least squares technique that minimizes distance between corresponding points in the model and the observation. Though their system demonstrates superior performance, they take 2 minutes to process a single frame. Their system is extended to estimate finer surface deformation [3], for use with multiple subjects [50] and for use with moving unsynchronized cameras [94].

Sigal, Balan and Black [15] perform a quantitative evaluation and report

CHAPTER 2. LITERATURE SURVEY

that APF as proposed in [53] is able to track successfully with 1000 particles without using any temporal motion prior. Though this is promising, even with 1000 particles, processing time per frame when realized in MATLAB is reported to be 240 seconds [15] for the tracking method that produce best results. Sigal, Balan and Black [49], extend the HMC system to use a statistical model of the human subject [84]. In [96], they use shading as an additional cue for pose estimation by constructing a cost function that exploits the knowledge of light sources. An integrated discriminative and generative method is presented in [51] for initialization of shape and pose as well as tracking. Their system is extended to estimate the pose and shape of an arbitrary human subject from a single image in [97].

Pons-Moll et al. [95] fuse visual data with IMU (Inertial Measurement Unit) sensors that provide orientation information for the human joints to achieve human motion capture in outdoor scenario. The noisy data provided by IMU sensors are used to constrain the samples generated during global optimization [92, 26] using inverse kinematics which result in a more informative set of samples. Figure 2.5 shows the set of sample poses generated during global optimization with and without fusing the input from IMU sensors. It can be observed that when fusing input from the IMU sensors, the sample poses are more informative. However, their method is not a marker-less approach since they use IMU sensors.

Brubaker, Fleet and Hertzmann [98] present a low dimensional model for tracking walking persons. They rely on a physics based dynamic model coupled with a sequential Monte Carlo filter using an image based likelihood. Their approach is restricted to tracking walking people.

Riusa et al. [99] present an action specific dynamic model for human pose tracking using particle filters. They use monocular input for tracking and show tracking results for walking sequence.

2.3.4 Dimensionality reduction

Urtasun, Fleet, and Fua [100] present an approach to human pose tracking using Gaussian process dynamic model (GPDM) [101]. The method uses inspiration from Gaussian process latent variable model (GPLVM) [102], which is a non-linear dimensionality reduction technique. GPDM learns a

2.3. PREVIOUS WORK IN MODEL BASED MOTION CAPTURE

Figure 2.6: The figure shows the latent subspace learned with GPLVM [100] for golf swing (a), (b) and walking sequence (c), (d). Several different trajectories of golf swing and walking motion can be observed in (a) and (c). Density of the samples in the latent subspace can be observed in (b) and (d). It can be noticed that (a) is acyclic, whereas (c) is cyclic. This is caused by the acyclic and cyclic nature of golf swing and walking motion.

latent subspace and a dynamic prior in the latent subspace which results in typical motion such as golf swing and walking. They show that a latent subspace of dimension 3 captures most stylistic variations in a typical motion in a state-space with as many as 80 degrees of freedom. Figure 2.6 shows the latent subspaces learned for walking and golf swing analysis. Their method produces successful results even with monocular input using the dynamic model in the latent subspace as a prior. However, by relying on strong prior their technique restricts the type of human motion that can be captured.

2.3.5 Global optimization techniques from evolutionary computing

John et al. [103] present an approach to human motion capture using hierarchical particle swarm optimization. Particle swarm optimization (PSO) is a global optimization technique that was discovered as a result of modeling swarm intelligence [104]. It uses a set of particles each traveling at a velocity to systematically search the state space. The particles are attracted to the global optimum observed by all the particles and the local optimum observed by each particle in every iteration. This is achieved by adapting the particle velocity vectors in each iteration with an element of randomness. In [103], authors use PSO to optimize a high dimensional non-convex function similar to Eq. (2.1). In addition, they perform a hierarchical decomposition and search the state space of the human model hierarchically. Oikonomidis, Kyriazis and Argyros [105, 106] apply PSO for model based articulated hand

CHAPTER 2. LITERATURE SURVEY

tracking, which is a closely related problem. Zhao et al. [107] present an approach to human pose tracking using hierarchically annealed genetic algorithm. Their method learns a linear low dimensional subspace using PCA and performs optimization in the low dimensional subspace.

2.3.6 Other related methods

Drummond and Cipolla [24] model the subject to be composed of rigid quadric objects, each with 6 degrees of freedom. They use a pseudo-generative model and project the rigid quadric object to get a parametric conic that can be sampled from. These samples can be used to evaluate a function similar to equation (2.1). The statistics thus estimated for each rigid object is propagated along the kinematic tree to estimate the parameters that best explain change in pose temporally and spatially. Their method though novel is restricted by the maximum change in pose that can be handled.

Kohli et al. [108] present an integrated approach to human pose estimation and segmentation. They use a graph cut [109] based technique to segment the subject from the background. An appearance based likelihood is fused with the shape prior (obtained by projecting the model from previous time instance) to segment the subject from the background.

Carsten Stoll et al. [110] present a high speed approach to marker less motion capture. Their method borrows ideas from Pfinder [78]. Though their system operates at high speed, they make limiting assumptions such as constant appearance to achieve this.

2.4 Summary

In this chapter, we present a taxonomy of the current approaches to human pose tracking. Model free approaches, model based approaches and discriminative approaches were discussed. The main drawback in model free approach is the poor reconstruction accuracy due to the lack of a rigid skeleton. On the other hand, discriminative approaches require a large number of training samples and do not generalize to unknown motion. As a result,

2.4. SUMMARY

a model-based approach is pursued in this thesis. Prominent studies on model-based approach in the past two decades were discussed in depth.

The main scientific problem with model-based approach was identified to be optimizing a non-convex cost function constructed in a high dimensional state space. This results in high computational overhead for motion reconstruction. Hence there is a need to construct efficient techniques for optimization. There is also a need to formulate effective cost functions that incur a low overhead. Furthermore, methods that can reduce the dimensionality of the state space needs to be explored. The next chapter furnishes various details of the model-based approach to enable the rest of the thesis to address the above-mentioned problems.

CHAPTER 2. LITERATURE SURVEY

Chapter 3

Model-based method for human motion capture

This chapter gives a detailed formulation of generative model based pose tracking. In Section 2.1.1, pose estimation was defined as an optimization problem using equation (2.1). However, key details were not discussed including the details about the projection and the disparity function. Furthermore, the optimization view which requires exhaustive global optimization is not always feasible. Hence, some studies approach tracking as a filtering problem, i.e., estimate the pose with the uncertainty in it and propagate the estimate and the uncertainty temporally. In this chapter, human pose tracking is formulated as a filtering problem and the aspects that were not included in last chapter are discussed. This chapter also introduces terminologies that would be referred to in future chapters.

Figure 3.1: Temporal propagation graphical model. X_t represents the hidden state and Y_t represents the observed variable at time instance t .

3.1 Filtering formulation

In this section, the filtering formulation is derived, similar derivations can be found in [87, 27]. Let $X_t \in \mathbb{R}^d$ define the state of the model and Y_t represent a set of images observed at time instance t . Y_t would have C image frames where C is the number of cameras used in the system. X_t and Y_t would be referred to as the *State* and the *Observation* in the future. Let $\mathcal{X}_t$ and $\mathcal{Y}_t$ refer to the set of states $\{X_1, \dots, X_{t-1}, X_t\}$ and observations $\{Y_1, \dots, Y_{t-1}, Y_t\}$ till time t starting from the first frame. The problem of pose tracking can be posed as a problem of state estimation at time t having observed $\mathcal{Y}_t$. Formally, this can be expressed as estimating the distribution $P(X_t|\mathcal{Y}_t)$, which can be expressed as

$$P(X_t|\mathcal{Y}_t) \propto P(Y_t|X_t, \mathcal{Y}_{t-1})P(X_t|\mathcal{Y}_{t-1}) \quad (3.1)$$

Proportionality results from the application of special form of Bayes rule expressed below

$$P(a|b, c) = \frac{P(b|a, c)P(a|c)}{P(b|c)} \propto P(b|a, c)P(a|c) \quad (3.2)$$

Equation (3.2) can be proved by taking the denominator of RHS to the LHS and multiplying both sides by $P(c)$, then it can be shown that both sides are different expansions of $P(a, b, c)$ by applying chain rule. Equation (3.1) follows from applying (3.2) by considering random variate a to be X_t , b to be Y_t and c to be $\mathcal{Y}_{t-1}$. Equation (3.1) can be simplified further but it requires few assumptions related to the graphical model [71] shown in 3.1. This is a well-known model used for temporal propagation of state information. The assumptions are listed below

- The state space model is assumed to be a first order Markov model i.e $P(X_t|\mathcal{X}_{t-1}) = P(X_t|X_{t-1})$
- The observations are assumed to be independent of each other and the process i.e $P(Y_t|\mathcal{X}_t, \mathcal{Y}_{t-1}) = P(Y_t|X_t)$

Second part of the RHS in equation (3.1) can be expressed as below

$$P(X_t|\mathcal{Y}_{t-1}) = \int_{d\mathcal{X}_{t-1}} P(\mathcal{X}_t|\mathcal{Y}_{t-1}) \quad (3.3)$$

3.1. FILTERING FORMULATION

Applying chain rule

$$P(X_t|\mathcal{Y}_{t-1}) = \int_{d\mathcal{X}_{t-1}} P(X_t|\mathcal{X}_{t-1}, \mathcal{Y}_{t-1})P(\mathcal{X}_{t-1}|\mathcal{Y}_{t-1}) \quad (3.4)$$

Applying the assumptions for temporal propagation

$$P(X_t|\mathcal{Y}_{t-1}) = \int_{d\mathcal{X}_{t-1}} \int_{d\mathcal{X}_{t-2}} P(X_t|X_{t-1})P(\mathcal{X}_{t-1}|\mathcal{Y}_{t-1}) \quad (3.5)$$

Marginalizing $\mathcal{X}_{t-2}$ would result in

$$P(X_t|\mathcal{Y}_{t-1}) = \int_{d\mathcal{X}_{t-1}} P(X_t|X_{t-1})P(X_{t-1}|\mathcal{Y}_{t-1}) \quad (3.6)$$

Applying the assumptions for temporal propagation to the first part of the RHS in equation (3.1) we can show

$$P(Y_t|X_t, \mathcal{Y}_{t-1}) = P(Y_t|X_t) \quad (3.7)$$

Substituting equation (3.7) and equation (3.6) into equation (3.1) we get

$$P(X_t|\mathcal{Y}_t) \propto P(Y_t|X_t) \int_{d\mathcal{X}_{t-1}} P(X_t|X_{t-1})P(X_{t-1}|\mathcal{Y}_{t-1}) \quad (3.8)$$

Equation (3.8) is a critical result in filtering formulation. It is of the form

$$Posterior(t) \propto Likelihood * Prior \quad (3.9)$$

$$Prior = TemporalDynamics \otimes Posterior(t-1) \quad (3.10)$$

where *Likelihood* $P(Y_t|X_t)$ is the unnormalized probability of observing Y_t as a function of the underlying state X_t and *Prior* $P(X_t|\mathcal{Y}_{t-1})$ is the predicted probability of the state X_t having observed $\mathcal{Y}_{t-1}$ and the *Posterior* $P(X_t|\mathcal{Y}_t)$ is the probability of the state X_t having observed $\mathcal{Y}_t$. Prior can be obtained by convolving *TemporalDynamics* $P(X_t|X_{t-1})$ with the *Posterior* from last time instance. Being a probability density function, exact Posterior can be obtained by normalizing RHS such that it integrates to 1.

This formulation of state estimation is very generic. It does not constrain the distribution to be of any specific form. Hence this formulation can be customized by making specific choices about the representation of the

CHAPTER 3. MODEL-BASED METHOD FOR HUMAN MOTION CAPTURE

distribution, likelihood function and temporal dynamics used by the system.

3.2 Optimization vs filtering

In last chapter the optimization view (2.1) of human pose tracking was described. This can be related to the filtering view. The negative log of the *Likelihood* function described in last section is equivalent to the disparity function *diff* defined in (2.1). A tracking algorithm can ignore past estimates of the state and exhaustively search for the best estimate for the current time instance, i.e., it simply obtains the Maximum Likelihood Estimate (MLE)

$$\hat{X}_t = \arg \max_{X_t} P(Y_t | X_t) \quad (3.11)$$

However, since the underlying state does not change drastically in a single time instance, considering the prior estimate of the state would make the tracking algorithm robust and efficient. This is achieved by estimating the posterior distribution for every discrete time-step.

3.3 Human shape and pose

In this section, we describe the representation of the pose and shape parameters of the human, since it enables the exposition of other aspects such as the likelihood. Most studies in the current literature model humans as a rigid articulated object. Kinematics is used to model articulation. Human limbs are commonly represented by cylinders [27], superquadric ellipsoids [22], truncated cones [26, 15]. Hence, we stick to this representation in this section. However, modern work in human motion capture, model humans exactly using statistical models [84, 85, 49, 111, 97] or deformable surface meshes [95, 50]. Figure 3.2 shows the truncated cone that is used to represent all body parts including torso, head, upper arm, lower arm, upper legs and lower legs.

3.3. HUMAN SHAPE AND POSE

Figure 3.2: A human limb represented by truncated cone. The figure shows the side view and the cross section of the truncated cone with elliptical cross section. The parameters (L , D_B , D_E , C_R) that define the limb are also shown.

3.3.1 Shape

The parameters (L , D_B , D_E , C_R) shown in Figure 3.2 control the shape of the human parts. Commonly 10 to 15 such truncated cones are used to represent a complete human body, a 10 part representation would have 4 parts for the left and right hand, 4 for left and right leg, 1 for torso and 1 for the human head. The parameters (L , D_B , D_E , C_R) for all the parts of the human body is collectively referred to as shape parameters [10].

Assuming a study uses 10 such parts to model a human there are 40 parameters that define the shape. However in practice the underlying degrees of freedom of these shape parameters are far less than 40. A number of assumptions are made about the human body that constrain and thus reduce the dimensionality of the shape parameter space. Common assumption made in most study is symmetry between the left and right half of the human body. Some research studies even impose anthropometric constraints between ratios of limb lengths [10].

3.3.2 Pose

Figure 3.2 shows a limb defined in its own reference frame. A rigid body in $\mathbb{R}^3$ has 6 degrees of freedom for translation (3) and rotation (3). Therefore,

CHAPTER 3. MODEL-BASED METHOD FOR HUMAN MOTION CAPTURE

a naive way to represent a human pose in $\mathbb{R}^3$ would be to use 6 parameters for each limb which would add up to 60 for the 10 part representation. However, in practice, far less than 60 is needed due to kinematic constraints between limbs. When the position of the torso is given, the location of the shoulder cannot be arbitrary, it is constrained to be at a fixed offset from the torso. Hence the translation parameters for the shoulder joint would be a constant vector. Furthermore, all the human joints do not have the same degree of freedom. It varies depending on the joint. For example, a shoulder joint has three degrees of freedom while a knee joint has only one. Such constraints further reduce the dimensionality of the space needed to represent the human pose. Recent research use between 25 to 50 parameters to represent a human pose.

3.3.3 Kinematic tree

The most common method used to construct the kinematic model in various study would be using a homogeneous matrix [27] as shown below

$$T = \begin{pmatrix} R_z R_y R_x & \tau \\ \mathbf{0} & 1 \end{pmatrix} \quad (3.12)$$

where $R_x, R_y, R_z \in \mathbf{SO}(3)$ represent rotation along x, y and z axis by θ_x, θ_y and θ_z respectively and τ represent the translation vector in $\mathbb{R}^3$, $\mathbf{0}$ is a row vector in $\mathbb{R}^3$ and 1 is a scalar constant. Using homogeneous matrices, a kinematic tree could be constructed that can assemble all the limbs in global coordinates as a function of the pose parameters.

A typical kinematic tree with 10 parts would have the torso in the root of the tree and all parts connected to the torso like the head, upper left hand, upper right hand, upper left leg and upper right leg would be its immediate children. Furthermore, lower left hand, lower right hand, lower left leg and lower right leg would be the immediate children of upper left hand, upper right hand, upper left leg and upper right leg respectively. Given that each part is defined in its own local reference frame which can be expressed in its parent's reference frame by left multiplication with the homogeneous matrix T_l , any part's local reference frame can be expressed in global coordinates by recursively multiplying the homogeneous matrices of the part along with

3.4. LIKELIHOOD

all its parents. This can be formally expressed as below

$$\forall l \in \text{parts}, \forall x \in S(l), G_l^x = \left(\prod_{i \in l, \text{Par}(l)} T_i \right) x \quad (3.13)$$

where *parts* represent the set of parts used to represent the complete human, the operator $S(l)$ generates the set of points in a truncated cone with elliptical cross section parametrized by the shape parameters of the part l , operator $\text{Par}(l)$ returns the ordered list of parents of part l in the kinematic tree. The above definition assumes that the product recursion is done in order (part, parts immediate parent and so on) from right to left since matrix multiplication is not commutative. Furthermore, the homogeneous matrix T_l corresponding to the root of the kinematic tree expresses the local reference of the torso with respect to the global coordinates.

It can be observed that the shape parameters are encoded in the operator $S(l)$, where as the pose parameters are encoded in the matrix T_l . Depending upon the part and the joint type (degrees of freedom of the joint) one or more of the angles that make T_l , (θ_x , θ_y and θ_z) would remain constant, while the translation parameter for all parts except the root of the kinematic tree (torso) would remain constant. Therefore, a complete human pose is represented by the global translation parameter τ and the set of angles that are non-constant. This is usually represented as a vector $X \in \mathbb{R}^d$ where d is the degrees of freedom of the human pose.

3.4 Likelihood

In this section, the filtering formulation is furnished with the details of the likelihood function. As described in Section 3.1, the function measures the likelihood of observing Y for a given value of X . In this section, we define a relatively simple likelihood function using silhouettes. However, this is not the only way in which likelihood function can be defined. A formulation that models the appearance of each part is described in [27]. Furthermore, we define an image based likelihood; more sophisticated voxel based likelihoods are used in some studies [86].

Let C be the number of cameras used in the HMC system. A set of

CHAPTER 3. MODEL-BASED METHOD FOR HUMAN MOTION CAPTURE

Figure 3.3: Likelihood generation explanation [15](best seen in color) a) Shows the silhouette R_t b) Shows the synthetic projection Z_X c) Shows the superimposition of the two

synthetic silhouettes B_X^i , $i \in \{1, \dots, C\}$ can be generated as a function of the pose parameter X , by projecting the model to virtual cameras with same intrinsic and extrinsic parameters [81] as the multi-view setup. Let R_t^i , $i \in \{1, \dots, C\}$ be the silhouettes obtained after background subtraction from the image *Observation* Y_t . Then, the log likelihood can be defined as below

$$\log P(Y_t|X_t) = - \sum_{i=1}^C \text{diff}(B_X^i, R_t^i) \quad (3.14)$$

where diff defined below is a measure of disparity between the two images.

$$\text{diff}(B, R) = \alpha I_t + (1 - \alpha) O_t \quad (3.15)$$

$$I_t = \frac{1}{|\text{ind} \in R, R(\text{ind}) \neq 0|} \sum_{\text{ind} \in B}^{B(\text{ind})=0} R(\text{ind}) \quad (3.16)$$

$$O_t = \frac{1}{|\text{ind} \in B, B(\text{ind}) \neq 0|} \sum_{\text{ind} \in B}^{B(\text{ind}) \neq 0} 1 - R(\text{ind}) \quad (3.17)$$

The above formulation of the likelihood can be best explained with the Figure 3.3.

The figure shows the silhouette R_t obtained by background subtrac-

3.5. TEMPORAL DYNAMICS

tion from the *Observation* in red. The synthetic silhouette B_X obtained by projection is shown in blue. As described earlier *diff* is a measure of disparity between the two binary images and hence is a minimization objective. A good likelihood function would maximize the yellow region in the superimposed image which is achieved by O_t in Eq. (3.17). Since *diff* is a minimization objective, O_t is defined as a function that minimizes the set of blue pixels in the superimposed image as a fraction of all blue pixels in B_X , which is the same as maximizing the yellow region. Also, a good likelihood function would minimize the red region in the superimposed image, without this a pose which completely fits inside the silhouette for example by moving the model away from the camera in perpendicular direction would be favored. This is prevented by the function I_t Eq. (3.16). I_t minimizes the red pixels in the superimposed image as a normalized fraction of all red pixels in R_t .

Equations (3.16) and (3.17) assume that the foreground (human) is represented by 1 and the background is represented as 0 in both R_t and B_X . The parameter α in Eq. (3.15) decides the relative importance of the two objectives. Generally, both O_t and I_t are treated as equally important (i.e., $\alpha = 0.5$).

3.5 Temporal dynamics

This section furnishes information about temporal dynamics $P(X_t|X_{t-1})$, which is often used by the inference algorithm to be more efficient. Intuitively, the term helps to predict the diffusion undergone by the previous posterior distribution in a time instance. The term is commonly modeled by a stochastic difference equation. Often the state space is expanded to include previous states to represent second or higher order difference equation. We assume all model to be of Gaussian uncertainty since this is by far the most common distribution used. The covariance matrix Σ (Σ is a symmetric positive definite matrix) can be either learned from data or can be determined empirically. We use the notation $\mathcal{N}(x; \mu, \Sigma)$ to represent a Gaussian with mean μ and covariance Σ .

The simplest model that could be used is a constant position model,

 CHAPTER 3. MODEL-BASED METHOD FOR HUMAN MOTION CAPTURE

which can be formally expressed as below

$$P(X_t|X_{t-1}) = \mathcal{N}(X_t; X_{t-1}, \Sigma) \quad (3.18)$$

which simply states that the distribution of the next state is a Gaussian centered around the previous state.

A more general model would be a constant velocity model [12] which can be formally expressed as below

$$P(X_t|X_{t-1}) = \mathcal{N}(X_t; 2X_{t-1} - X_{t-2}, \Sigma) \quad (3.19)$$

Though the above equation violates the first order Markov assumption, it works well in practice.

An even more sophisticated method would be to model dynamics by a linear state transition matrix as below

$$P(x_t|x_{t-1}) = \mathcal{N}(x_t; Ax_{t-1}, \Sigma) \quad (3.20)$$

The parameters A and Σ can be learned from training data [87] specific to an action using techniques such as Maximum Likelihood Estimates (MLE). Equation (3.20) uses x_t rather than X_t to indicate that the state space could be extended to include past states.

Apart from the above three which are linear, other nonlinear state transition models can also be used which can be summarized as below

$$P(x_t|x_{t-1}) = \mathcal{N}(x_t; f(x_{t-1}), \Sigma) \quad (3.21)$$

where f is a non-linear vector valued function of vector input of the same dimension as the state space $\mathbb{R}^d$. Some studies [23] have resorted to implicit modeling of dynamics rather than explicitly modeling it with stochastic difference equations.

3.6 Bayesian Inference

In this section the filtering formulation is extended further and various Bayesian inference methods such as Kalman filters, Multiple Hypothesis Trackers and Particle filters are considered in this framework. Their performance and reasons for failure with human pose tracking is analyzed.

3.6.1 Kalman filter

Kalman filter [112, 113] makes the following 2 assumptions

- Temporal dynamics is a linear model with Gaussian uncertainty similar to equation (3.20)
- Likelihood function is unimodal and can be approximated by a Gaussian

These two assumptions allow the prior and the posterior to be a parametric Gaussian and thus enable a closed form solution for tracking.

Complying with the above 2 assumptions the posterior for time $k-1$, temporal dynamics and the likelihood can be expressed

$$P(X_{k-1}|\mathcal{Y}_{k-1}) = \mathcal{N}(X_{k-1}; \hat{X}_{k-1}, P_{k-1}) \quad (3.22)$$

$$P(X_k|X_{k-1}) = \mathcal{N}(X_k; AX_{k-1}, Q) \quad (3.23)$$

$$P(Y_k|X_k) = \mathcal{N}(Y_k; HX_k, R) \quad (3.24)$$

Where $X_* \in \mathbb{R}^x, Y_* \in \mathbb{R}^y$ are the state and observation vectors of dimensionality x and y respectively. $P_*, Q \in \mathbb{S}_+^x, R \in \mathbb{S}_+^y$ are symmetric positive semi-definite covariance matrices. A and H are $x \times x$ and $y \times x$ matrices respectively. Substituting the terms into the filtering equation (3.8) results in the form

$$P(X_k|\mathcal{Y}_k) \propto \mathcal{N}(Y_k; HX_k, R) \int_{X_{k-1}} \mathcal{N}(X_k; AX_{k-1}, Q) \mathcal{N}(X_{k-1}; \hat{X}_{k-1}, P_{k-1}) \quad (3.25)$$

CHAPTER 3. MODEL-BASED METHOD FOR HUMAN MOTION CAPTURE

It can be shown that the convolution results in a Gaussian prior density which is equivalent to the Kalman predict equations below

$$P(X_k | \mathcal{Y}_{k-1}) = \mathcal{N}(X_k; X_k^-, P_k^-) \quad (3.26)$$

$$X_k^- = A\hat{X}_{k-1} \quad (3.27)$$

$$P_k^- = AP_{k-1}A^T + Q \quad (3.28)$$

where X_k^- and P_k^- refer to the mean and variance, respectively, of the Gaussian prior density. When the prior is multiplied by the likelihood conditioned by the actual observation Y_k , it results in a unnormalized Gaussian posterior density of dimension x . It can be shown [114] that the parameters of the posterior density can be estimated by maximizing the log posterior in closed form and such an attempt would result in the Kalman update equations below

$$\hat{X}_k = X_k^- + K_k(Y_k - HX_k^-) \quad (3.29)$$

$$P_k = (I - K_kH)P_k^- \quad (3.30)$$

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \quad (3.31)$$

Figure 3.4 shows the changes undergone by the posterior in one time frame by Kalman filtering. Graph in the top left corner shows the posterior in last time instance as a 1D Gaussian. The deterministic part of the stochastic difference equation makes the distribution undergo a deterministic drift as shown in the top right graph. This is the state transition induced by the matrix A in Equation 3.23. The stochastic part of the difference equation (resulting from the Gaussian with covariance Q in Equation 3.23) has the effect of increasing the uncertainty which is shown in the bottom right corner. The graph in the bottom right corner represents the prior density expressed in Equation 3.26. The measurement induces a Gaussian likelihood on the state. The mean of the Gaussian likelihood is represented by the arrow in the bottom left graph. Multiplication of the likelihood and the prior density results in a posterior Gaussian density shown in the bottom left graph of Figure 3.4.

Kalman filters have been a very effective tool and have been applied to a number of problems. However, they do not function well with human

3.6. BAYESIAN INFERENCE

Figure 3.4: Intuitive demonstration of Kalman filters [87]. The figure shows the changes undergone by the posterior from last time instance as a consequence of one Kalman filtering iteration.

pose tracking [26, 87]. The main reason for this is, the likelihood function is multi-modal due to the nonlinearities caused by joint angles in a human pose. Attempting to approximate such multi-modal distribution with a Gaussian distribution results in catastrophic failure [26].

3.6.2 Multiple hypothesis trackers

Multiple hypothesis tracking [22, 25] is a technique that improves on Kalman filters. As described in previous section, the main reason for the failure of Kalman filters when applied to human pose tracking is that it assumes the measurement uncertainty to be uni-modal. However, in practice it is multi-

CHAPTER 3. MODEL-BASED METHOD FOR HUMAN MOTION CAPTURE

No	Steps undertaken in Multiple Hypothesis Tracking
1	Each hypothesis (mode) in the posterior from last time instance is processed with the Kalman predict equation to get the prior
2	Samples are drawn from the prior distribution
3	Samples from prior distribution are locally optimized in the likelihood function to estimate the local modes in it
4	The local modes in likelihood function are expressed in parametric form
5	Likelihood function is multiplied by the prior to obtain the posterior in parametric form

Table 3.1: Steps undertaken in a Multiple Hypothesis Tracker.

modal. Multiple Hypothesis Trackers works around this by maintaining a bank of Kalman filters that can track arbitrary multi-modal distribution. The steps undertaken in a Multiple Hypothesis Tracker is shown in Table 3.1

A multiple hypothesis tracker relies on a parametric form that can represent a complex multi-modal distribution to represent the prior, likelihood and the posterior. Individual components or modes that makeup the whole distribution is referred to as hypothesis. A critical choice to be made with such a tracker would be the parametric representation itself. Mixture of Gaussian would be the natural choice [22] for the representation. However Cham and Rehg [25] rely on novel distributions like Piecewise Gaussian (PWG) in order to avoid the high fitting cost incurred by EM (to fit a MoG). Another critical part is the method used to multiply the likelihood with the prior. Since each hypothesis is a Gaussian they can be multiplied parametrically to produce another Gaussian[115], however there can be an explosive number of terms in such a multiplication. Hence the most compatible prior and likelihood hypotheses alone are multiplied to form a posterior hypothesis.

Though multiple hypothesis tracking overcomes the problems in Kalman filter, it's main drawback is it's high computational overhead. It depends on local optimization techniques which in turn rely on analytic expression for first and second derivatives of the function *diff* in Eq. (3.15). This severely restricts the choice of likelihood. Alternatively, if a numerical technique is

3.6. BAYESIAN INFERENCE

resorted to, the number of evaluation of the function is high for computing gradient and Hessian. Hence this method incurs significant computational overhead.

3.6.3 Particle filters

Particle filtering [87] is a very successful and a relatively simple technique for filtering. The filtering formulation, iteratively estimates the posterior distribution. The posterior density is a function of same dimensionality as the state space. In practice estimates that summarize the distribution is more useful than the distribution itself. There are two such estimates which are most often used, they are listed below

$$MAP = \arg \max_{X_t} P(X_t | \mathcal{Y}_t) \quad (3.32)$$

$$E[x] = \int_{dX_t} x P(X_t | \mathcal{Y}_t) \quad (3.33)$$

Monte Carlo [116] methods show that both MAP and the expected state can be calculated with a set of samples drawn from the distribution, also the variance of such an estimate asymptotically converges to the true value as the number of samples approach ∞ . Furthermore, using techniques such as factored sampling [87], all the steps in the filtering formulation can be done using a set of samples. Thus a set of samples are sufficient to represent the prior, likelihood and the posterior non-parametrically.

Figure 3.5 captures the steps in particle filtering. Particle filters represent the posterior by a set of fixed number of samples and their weights which sum to 1 as shown below

$$(s_k^i, w_k^i), i \in \{1, \dots, N\} \quad (3.34)$$

$$P(X_k | \mathcal{Y}_k) = \sum_{i=1}^N w_k^i \delta(X_k - s_k^i) \quad (3.35)$$

Subscript k is used to represent time-step t to be relevant with respect to Figure 3.5. First row of the Figure 3.5 shows the weighted set of samples that represent the posterior density from time-step k-1. Each sample is

CHAPTER 3. MODEL-BASED METHOD FOR HUMAN MOTION CAPTURE

Figure 3.5: Intuitive demonstration of particle filters [87]. The figure shows the changes undergone by the posterior from last time instance as a consequence of one particle filtering iteration

represented by a circle in the state space location s_k^i which is 1D in the figure. The radius of the circle is proportional to its weight w_k^i . At the start of a filtering iteration, the set of samples are re-sampled with replacement to obtain N samples. This procedure is used to generate samples that are of equal weight $1/N$. Then these samples are made to undergo deterministic drift that is caused by the stochastic difference equation. The samples in second row of the figure captures these two process. The sample weights are uniform as indicated by a fixed radius of the circle. Also it can be observed that not all samples survive the re-sampling process. The samples that had a high weight before resampling are selected more than once while some with less weight are not selected at all. It can also be observed that the drift is deterministic. When a sample with high weight is selected multiple times, all of them undergo the same drift as indicated by several arrows to a single sample in second row.

The third row captures the change caused by the stochastic part of the stochastic difference equation i.e. addition of noise. This process is referred to as diffusion. It can be observed from the figure that the samples that were

3.6. BAYESIAN INFERENCE

grouped together in previous stage spreads as a result of diffusion. The set of samples at the end of diffuse stage represent the prior distribution. This can be easily shown by considering the motion prior to be of the form in Eq. (3.20).

$$P(X_k|X_{k-1}) = \mathcal{N}(X_k; AX_{k-1}, Q) \quad (3.36)$$

$$P(X_k|\mathcal{Y}_{k-1}) = \int P(X_k|X_{k-1})P(X_{k-1}|\mathcal{Y}_{k-1})dX_{k-1} \quad (3.37)$$

$$= \int \mathcal{N}(X_k; AX_{k-1}, Q) \sum_{i=1}^N w_{k-1}^i \delta(X_{k-1} - s_{k-1}^i) dX_{k-1} \quad (3.38)$$

$$= \sum_{i=1}^N w_{k-1}^i \int \mathcal{N}(X_k; AX_{k-1}, Q) \delta(X_{k-1} - s_{k-1}^i) dX_{k-1} \quad (3.39)$$

$$= \sum_{i=1}^N w_{k-1}^i \mathcal{N}(X_k; As_{k-1}^i, Q) \quad (3.40)$$

Thus the resulting density can be considered to be a Mixture of Gaussian (MoG) or a kernel density. Generating N samples from the MoG is exactly equivalent to the three steps done above namely re-sampling, drift and diffuse.

The prior density should be multiplied by the likelihood to obtain the posterior for the current iteration. This is achieved by simply evaluating the likelihood function at the samples and re-normalizing the weights such that they sum to 1. Last step can be better explained by considering the prior distribution as a proposal distribution for importance sampling [116] from the likelihood. Formally this can be expressed as below

$$P(X_k|\mathcal{Y}_{k-1}) = \sum_{i=1}^N \frac{1}{N} \delta(X_k - s_k^i) \quad (3.41)$$

$$P(Y_k|X_k) = \sum_{i=1}^N \frac{w_k^i}{1/N} \delta(X_k - s_k^i) \quad (3.42)$$

where w_k^i is the normalized weight after evaluating the likelihood function at the prior samples. Division by $1/N$ is required since prior distribution is used as a proposal distribution for sampling from the likelihood. When the prior density is multiplied by the likelihood we obtain the posterior density

CHAPTER 3. MODEL-BASED METHOD FOR HUMAN MOTION CAPTURE

as below.

$$P(X_k|\mathcal{Y}_k) = \sum_{i=1}^N \frac{1}{N} \frac{w_k^i}{1/N} \delta(X_k - s_k^i) = \sum_{i=1}^N w_k^i \delta(X_k - s_k^i) \quad (3.43)$$

The above equation shows that considering the particles (s_k^i, w_k^i) to be drawn from the posterior distribution is theoretically well-founded.

This survey was primarily restricted to human pose tracking. Particle filters, which is also referred to as sequential Monte Carlo (SMC) methods, have been successfully applied to numerous other problems in computer vision as well as in other areas like signal processing and statistics. Readers are referred to [117] for a detailed description of particle filters and application to other problems. Particle filters have been applied to pose tracking [87, 26, 27] with limited success. The number of particles required to track human pose has been found to be unacceptably high (10^4). Theoretical analysis [118] attributes the reason to high dimensionality of the human pose and the ineffectiveness of importance sampling in such high dimensional spaces.

3.7 Global Optimization

In this section, we discuss various global optimization techniques that are used with model based methods for inference.

3.7.1 Annealed particle filter

Annealed Particle Filter (APF [26]) extends Particle filters with annealing [119]. The method was originally proposed for visual tracking of humans [53] in 2000 by J. Deutscher, A. Blake and I. Reid. However, it is a generic method that could be applied for stochastically searching for the Maximum Likelihood Estimate (MLE) of a high dimensional problem. Before describing APF a brief introduction to simulated annealing is provided.

3.7. GLOBAL OPTIMIZATION

Figure 3.6: Effect of annealing. The figure shows how annealing can change a distribution as the parameter T is varied from ∞ to 0. The distribution is a mixture of one dimensional unit variance Gaussian with weights (0.6, 0.4) and means (-4, 4).

3.7.1.1 Simulated annealing

Simulated Annealing [26] is a well known technique in optimization. It uses Markov Chain Monte Carlo (MCMC) based sampling method specifically Metropolis [116] algorithm to generate samples from an arbitrary high dimensional distribution. The sampling technique is combined with annealing to locate the global maximum of the distribution. The process of annealing can be expressed by the below equation

$$P(x) \propto e^{-\frac{U(x)}{T}} \quad (3.44)$$

where $U(x)$ is a minimization objective and $P(x)$ is a maximization objective that can be perceived as an unnormalized distribution since it is non-negative for all x . T (for Temperature) is a variable that can transform the distribution $P(x)$ from a uniform distribution to a delta function in the global maximum as it is varied from ∞ to 0. The effect of annealing a distribution can be seen in Figure 3.6. Simulated annealing uses Metropolis

CHAPTER 3. MODEL-BASED METHOD FOR HUMAN MOTION CAPTURE

[116] sampling method to generate a set of samples from the distribution $P(x)$, a symmetric proposal distribution (like Gaussian) is used to generate proposals which are accepted with the probability $\min\left(1, \frac{P(x)}{P(x')}\right)$ to create a random walk that is theoretically guaranteed to generate an independent set of samples from the distribution $P(x)$. The random walk starts from an arbitrary x_0 . However, such a sampling method would not work when T is 0, since the probability of landing in an exact location (the location of the delta function corresponding to the global maximum) is zero with any proposal distribution. Hence, simulated annealing uses a flatter distribution corresponding to higher values of T to start with, then the temperature is gradually reduced and new samples are generated from the last state. The sequence of temperatures is referred to as the annealing schedule which should be slow to avoid getting trapped in local minimum. However, slow annealing would incur high computational load, hence the annealing schedule is chosen as a trade-off between the two extremes.

3.7.1.2 APF overview

Annealed particle filter [26] incorporates annealing into the particle filtering framework. In this section an overview of APF is provided which is slightly different yet consistent with the way it was originally proposed [53]. Each particle filter iteration starts with a set of samples representing the posterior from last time instance and ends with a set of samples representing posterior in the current time instance. Annealed particle filter starts with a delta distribution in the global maximum of the likelihood from last time instance and iteratively constructs a set of samples that represent the likelihood for the current time instance at successively reduced annealing temperatures. Each iteration is exactly same as a particle filter iteration if the following assumptions are made

- Posterior distribution that is used by the first iteration of APF and that is produced by the last iteration of APF is a delta function in the global maximum of the likelihood.
- Temporal dynamics used in every iteration other than the first is a constant position model (Eq. 3.18) with a gradually reducing noise variance defined by the annealing schedule. First iteration could use

Figure 3.7: The figure demonstrates Annealed Particle Filtering. The MLE of a multi-modal likelihood shown by the graph is searched by APF using 5 layers. It can be observed that the samples in each layer shown by dark and light green circles, approximate an annealed version of the likelihood. The estimate of the optimum is shown in each layer by the red dot. It can be observed that the estimate slowly converges to the optimum.

any of the four temporal models defined in Section 3.5.

- The likelihood function in each iteration is an annealed version of the original likelihood obtained by gradually reducing T in equation (3.44) as per the annealing schedule.

Thus APF uses several particle filtering iterations to estimate the state having observed a new *Observation*. This iterative procedure combined with a good annealing schedule ensures that the samples in the last iteration of APF would represent the delta function in the global maximum as shown in the last plot of Figure 3.6. As mentioned before a single iteration of APF to process a new *Observation* consists of several inner iterations which will be referred to as layers in the future following the convention in [53]. Most implementations [15] of APF uses 5 layers with 200 particles in each layer thus resulting in 1000 likelihood evaluations, in other words 1000 likelihood evaluations are required per video frame to track a human with APF.

3.7.1.3 APF formal description

The Annealed Particle Filtering algorithm is described in detail here. We use Figure 3.7, which shows APF searching the global mode of a multi-modal likelihood to describe the procedure.

1. Each iteration starts with a point estimate of the state from the previous iteration. This can be considered as a delta distribution at the estimated state. Optionally, the delta distribution could be subject to a deterministic drift by a non-linear or a linear function f as below

$$s_0 = f(\hat{X}_{t-1}), w_0 = 1 \quad (3.45)$$

2. N samples are constructed in layer m (which runs from 1 to M), from the distribution of previous layer and are perturbed by Gaussian noise with variance B_m to result in a new set of samples

$$s_m^i = \text{sample}(s_{m-1}, w_{m-1}) + n \quad (3.46)$$

$$n \sim \mathcal{N}(0, B_m), \quad i \in \{1, \dots, N\}, \quad (3.47)$$

where $\text{sample}(s, w)$ is a function that generates a new sample from the set of weighted samples (s, w) . The samples thus obtained can be observed in Figure 3.7 in different layers, represented as light green circles with uniform radius.

3. The generated samples are weighted by evaluating the annealed likelihood function as below

$$w_m^i \propto \text{Likelihood}(s_m^i)^{\beta_m}, i \in (1, \dots, N) \quad (3.48)$$

$$\beta_m = \frac{1}{T_m} \quad (3.49)$$

$$\sum_{i=1}^N w_m^i = 1 \quad (3.50)$$

This results in a new set of samples (s_m, w_m) that will be used by the next layer (in 2). The set of weighted samples in different layers are observed in Figure 3.7, as dark green circles with the radius encoded

3.7. GLOBAL OPTIMIZATION

by the weights.

4. At the end of the last layer M , point estimate of the state is obtained as below

$$\hat{X}_t = \sum_{i=1}^N w_M^i s_M^i \quad (3.51)$$

The estimate of the state in each layer is observed in Figure 3.7 as red pentagrams. It can be observed that the estimates for weighted and uniform samples are different. This is due to the randomness introduced during resampling.

APF has been extended to include ideas from the Genetic Algorithm [120]. In [54], Deutscher et al. show that by using crossover operation, the performance of APF can be significantly improved. The extension selects $2N$ rather than N samples in step (2) above. Every sample pair is treated as parents (A and B) and a new sample (C) is generated for each pair by performing crossover. This is achieved by generating a random binary vector of same dimensionality as the state space. The elements of the vector C are chosen from parent A or B based on the value of the random vector being 0 or 1. This simple procedure is shown to produce 25% reduction in the number of samples needed to track in [54].

Annealed particle filter remains a very popular method in model based human tracking [15] since 2001 when it was proposed. Methods such as particle and Kalman filters are Bayesian methods that propagate the uncertainty in the estimate either using a Monte Carlo or a Gaussian approximation of the posterior. APF is not a Bayesian method, since it does not propagate the uncertainty in the estimate. It performs a stochastic search for the optimal state rather than filtering. However, it functions within our problem formulation and being a generic method it has been applied to other problems such as camera calibration [121].

Annealed particle filter has been extended in order to ensure certain theoretical results related to particle filters are relevant for it. The improved procedure is referred to as Interacting Simulating Annealing (ISA) [92]. As opposed to APF, ISA uses a static annealing schedule and a stratified resampling scheme. Gall et al. [14] show promising human motion capture results using the improved stochastic search procedure and an additional

CHAPTER 3. MODEL-BASED METHOD FOR HUMAN MOTION CAPTURE

Figure 3.8: The figure demonstrates Particle Swarm Optimization. The MLE of a multi-modal likelihood shown by the graph is searched by PSO using 5 layers. The particles in each layer are shown by light green circles. The estimate of the optimum is shown in each layer by the red dot. It can be observed that the estimate converges to the optimum.

local optimization step. The stochastic search procedure ISA shows modest improvement over APF.

3.7.2 Particle Swarm Optimization

Studies on articulated body motion capture have used other stochastic search methods such as particle swarm optimization [105, 106]. A brief description of this procedure is included here for completeness. PSO uses a set of particles similar to APF. Each particle has a position vector P_l^i and a velocity vector V_l^i , which are randomly initialized. Furthermore, each particle stores the best state observed during search G^i . The global best state observed by all the particles is also stored as a separate vector G . Once these variables are initialized, PSO explores the state space iteratively as below.

- For each particle i , the velocity vector is updated as

$$V_l^i = \omega V_l^i + \phi_p r_p (G^i - P_l^i) + \phi_g r_g (G - P_l^i)$$
- For each particle i , the position vector is updated as $P_l^i = P_l^i + V_l^i$
- Values of G^i and G are re-estimated.

In the exposition above, l is the iteration index, ϕ_p , ϕ_g and ω are free parameters. The parameters r_p and r_g are random variables drawn for each

3.8. SUMMARY

particle in every iteration, from a uniform distribution in the interval $[0, 1]$.

Figure 3.8 demonstrates the stochastic search operation using PSO on the same objective as before. Note that the objective searched by PSO is the negative log of the likelihood shown in the figure. The significant difference between APF and PSO is that, PSO does not aim to approximate the distribution with a set of particles as observed in Figure 3.8. For a high dimensional state space, APF is more popular as evident from the vast number of studies in the literature using it [49, 50, 15, 26, 12, 28, 122].

3.8 Summary

In this chapter, filtering formulation for human tracking was introduced. It was then compared with the optimization view introduced in last chapter. This was followed by specific examples of likelihood function and projection operation. Various temporal dynamic models used with this formulation was discussed. Finally, various inference algorithms used with the formulation was discussed in detail. We describe a novel inference method in next chapter.

CHAPTER 3. MODEL-BASED METHOD FOR HUMAN MOTION CAPTURE

Chapter 4

Stochastic Search

4.1 Introduction

In this chapter we extend our discussion on inference methods. We describe a novel stochastic search procedure referred to as parametric annealing (PA) for inference. We perform various study on PA such as scalability and sensitivity analysis. We further describe a systematic framework that can be used for parameter selection for the existing stochastic search methods described in last chapter as well as our novel method PA. We finally discuss how these methods can be extended for global optimization in the manifold $\mathbb{S}^n$.

4.2 Motivation

The APF algorithm for inference was described in detail in last chapter. The APF algorithm generates N new samples in each layer. However, samples from previous layers are discarded. Since the evaluation of likelihood incurs a significant overhead, one would want to retain the relevant samples from previous layers. The most natural way to do this would be to simply retain the samples x_i along with the corresponding likelihoods y_i , and raise them to a new annealing temperature T_m for the layer before normalization. However, this does not work well in practice.

CHAPTER 4. STOCHASTIC SEARCH

Figure 4.1: Test demonstrating effects of reusing samples. The subfigures a, b and c correspond to the configuration that discards samples, the one that retains them and the one that retains along with a deep annealing schedule. It can be observed that reusing makes the search less effective (b), which is overcome by augmenting it with a deep annealing schedule (c).

To demonstrate the effect of simply reusing, we conducted an experiment. We considered the log likelihood to be a quadratic, $-\frac{1}{2}x^T x$, where $x \in \mathbb{R}^{30}$. To verify how well APF searches through the state space we started from a fixed state and searched for the optimum. We compared two configurations of APF: the original configuration that discards samples, and a second configuration which retains them by raising the likelihood to the appropriate β_m for the layer (described in Eq. (3.49)) before normalization. To measure how effectively the methods locate the optimum, we obtained the distribution of the L2 norm of the state estimated by APF. The histograms of the L2 norm are shown in Figure 4.1 and 4.1. It can be observed that when samples are discarded, the state estimate is closer to the true optimum.

The test demonstrates the problems with retaining samples. The critical insight, however, is that the problems with reusing samples are more than overcome by choosing a “deep” annealing schedule that consistently reduces the annealing temperature, rather than a simple schedule which ensures 50% samples survive resampling. The precise formulation of a deep annealing schedule as a power law, which is a novel aspect of our method, is given in Section 4.3. The improvement is shown in the histogram in Figure 4.1 which augments retaining samples with our proposed deep annealing schedule. The result can be best explained by analyzing another simple example. Figure 4.2a shows a 1D Gaussian density in blue. A set of weighted samples were obtained by diffusing samples around $x = 3$. The samples are shown in red and the kernel density approximated by the samples using a Gaussian kernel

4.2. MOTIVATION

Figure 4.2: Example demonstrating the effect of annealing(color). It can be seen how annealing shifts the probability mass towards regions that improve the objective.

is shown in green. If the samples were to be re-sampled and diffused with the same Gaussian kernel, it is equivalent to generating samples from the density shown in green. Figure 4.2b shows the effect of annealing on the kernel density. It can be observed that annealing effectively shifts the kernel density mass towards regions that improve the search objective. Consequently, when enabled with a suitable annealing schedule, new samples are generated in areas of the state-space that improve the objective. Expressed differently, a good annealing schedule ensures that the worthless samples which end up in areas of the state-space with poor objective don't adversely impact the search process by quickly driving their normalized weights to zero in successive layers.

In addition, one would expect to have a slow annealing schedule when tracking articulated objects since their likelihoods are multimodal and fast annealing is known to converge to local optima. APF can be interpreted as generating new samples from a kernel density with a Gaussian kernel of gradually reducing noise covariance. It is well-known that the kernel density approximated by a set of samples is highly sensitive to the type of kernel. Therefore such a fixed diffusion schedule is very sensitive to the parameters,

CHAPTER 4. STOCHASTIC SEARCH

Figure 4.3: Example demonstrating the effect of a parametric form on the state estimate. The samples are shown in red, and an inferred parametric form is shown in green. The expected state of all samples, and the maximum of the parametric form, are shown in black and cyan respectively. It can be seen how the expected state is a worse estimate of the true optimum at -1.

more so when the annealing is slow and longer. This is evident from the two-fold reduction [54] in the number of samples needed to track when using an adaptive kernel. We take the idea of adaptive diffusion density a step further, and use an intermediate parametric form which is inferred from the set of samples to act as a substitute for the kernel density i.e. instead of resample and diffuse with a kernel, new samples are generated from a parametric model. The motivation is that, with an intermediate parametric form, the density would adapt even better.

Inferring an intermediate parametric form has an added advantage when the set of samples in the last layer of annealing still represent a multimodal density. Such a scenario can be observed in Figure 4.3, it shows a set of samples in the last layer of annealing in red. A parametric density inferred from the set of samples is shown in green. The parametric density shows that the samples represent a multimodal density. It can be observed that the expected state of such a sample set deviates from the global maximum, and a better estimate is obtained from the parametric form. These considerations motivate the need for an improved procedure, which we refer to as parametric annealing. We describe our method in detail in next section.

4.3. PARAMETRIC ANNEALING

4.3 Parametric Annealing

Let $P(x)$ ($x \in \mathbb{R}^d$) be a general multi-modal distribution, and let ψ be the set of N weighted samples (x_i, π_i) that represent $P(x)$. Formally,

$$P(x) \approx p(x; \psi) = \sum_{i=1}^N \pi_i \delta(x - x_i) \quad (4.1)$$

Let $q(x; \theta)$ be a Mixture of Gaussians (MoG), parametrized by θ .

$$q(x; \theta) = \sum_{c=1}^C \omega_c \mathcal{N}(x; \mu_c, \Sigma_c) \quad (4.2)$$

where C is the number of components, μ_c are the set of means, Σ_c are the set of covariances, ω_c are the set of weights for the MoG. They are collectively referred to as the parameter θ . We approximate the non-parametric model p by the parametric model q as follows.

$$q(x; \theta) \approx p(x; \psi) \approx P(x) \quad (4.3)$$

We estimate the parameters of $q(x; \theta)$ using Expectation Maximization (EM [123]) adapted to include the sample weights π_i . The EM algorithm associates a latent random variable z_i to each weighted sample (x_i, π_i) . The latent variable indicates the membership of the sample to one of the C clusters. Hence it is defined as the C valued categorical distribution $z_i \sim \text{Cat}(C, p)$. The probabilities for the latent variable are estimated in the E step as below

$$P(z_i = c) = \frac{\omega_c \mathcal{N}(x_i; \mu_c, \Sigma_c)}{\sum_{k=1}^C \omega_k \mathcal{N}(x_i; \mu_k, \Sigma_k)} \quad (4.4)$$

The M step of the EM algorithm estimates the parameters μ_c , Σ_c and ω_c using the current estimates for the random variable z_i as below.

$$\mu_c = \sum_{i=1}^N \pi_i P(z_i = c) x_i \quad (4.5)$$

$$\Sigma_c = \sum_{i=1}^N \pi_i P(z_i = c) (x_i - \mu_c)(x_i - \mu_c)^T + \xi \quad (4.6)$$

CHAPTER 4. STOCHASTIC SEARCH

$$\omega_c = \sum_{i=1}^N \pi_i P(z_i = c) \quad (4.7)$$

The parameter ξ is the regularizer described later in this section.

With this terminology, the high level steps done in each iteration of our algorithm are described in Algorithm 1.

Data: $\hat{x}_t$, the estimate of the state in last iteration

Result: Estimate of the state in this iteration

N^0 samples $x_i, i \in \{1, \dots, N^0\}$ are obtained by sampling from $\mathcal{N}(\hat{x}_t; \Sigma)$.

while $m \in \{1, \dots, M\}$ **do**

1. The likelihood y_i is evaluated (for example using Eq. (3.14)) for new samples x_i .
2. A suitable annealing temperature T_m is estimated using a deep annealing schedule.
3. Normalized sample weights $\pi_i^m = y_i^{\beta_m} / \sum_{i=1}^{N^m} y_i^{\beta_m}$, where $\beta_m = 1/T_m$ are estimated.
4. A parametric MoG approximation $q(x; \theta^m)$ is inferred from the weighted samples ψ using one EM iteration (single execution of the E and the M step).
5. C new samples x_i are generated from the parametric model and combined with old samples.

end

The global maximum is estimated from the parametric model.

Algorithm 1: Parametric Annealing

Though several steps involved in PA are common to APF (provided in Algorithm 2 for comparison), there are two novel aspects of PA. First, we reuse samples in step 5 in combination with a deep annealing schedule in step 2. Second, we fit a parametric model in steps 4 and 5 instead of using the kernel diffusion density, and obtain the final estimate from the parametric form in the final step.

The high level steps in Algorithm 1 include some of the parameters of the algorithm. The superscript m in any parameter indicates that it is a function of the layer, $m \in (1, \dots, M)$, where M is the number of annealing layers. The parameter N^m represents the total number of samples in layer m . As opposed to APF, where the number of samples in each layer remains fixed, the number of samples in PA grows in successive layers since we don't discard samples. As C new samples are introduced in each of M layer and

4.3. PARAMETRIC ANNEALING

Data: $\hat{x}_t$, the estimate of the state in last iteration

Result: Estimate of the state in this iteration

N samples $x_i, i \in \{1, \dots, N\}$ are obtained by sampling from $\mathcal{N}(\hat{x}_t; \Sigma)$.

while $m \in \{1, \dots, M\}$ **do**

1. The likelihood y_i is evaluated at sample x_i .
2. A suitable annealing temperature T_m that ensures only 50% samples will survive resampling is estimated.
3. Normalized sample weights are obtained. $\pi_i = y_i^{\beta_m} / \sum_{i=1}^N y_i^{\beta_m}$, where $\beta_m = 1/T_m$
4. N new samples are generated by sampling (resample and diffuse) from $\sum_{i=1}^N \pi_i \mathcal{N}(x_i; \alpha^m \Sigma)$.

end

The global maximum is estimated as the expected state of the weighted samples.

Algorithm 2: Annealed Particle Filter

we start with N^0 samples, the total number of samples for an iteration is $N^0 + MC$.

Each θ^m is the set of C means (μ_c^m), covariances (Σ_c^m), and mixture weights (ω_c^m), where c and m indicate that these parameters are dependent on the specific mixture component (c) and the layer (m). The parameter θ is made dependent on the layer m as a result of interleaving the EM algorithm with annealing. When EM is applied to determine the covariance, a regularizer ξ_m (see Eq. (4.6)) is added after the M step update for Σ_c^m to ensure that the covariance matrices are full rank. Since we are also annealing the samples while inferring the parametric form, we found that a gradually reducing regularizer is more effective than a fixed regularizer. Hence the regularizer is a function of the layer m .

The annealing schedule is an important part of our algorithm. We use a measure similar to particle survival rate α defined in [26] to control the annealing temperature:

$$\alpha(T_m) = \frac{1}{N \sum_{i=1}^N (\pi_i^m(T_m))^2} \quad (4.8)$$

$$\pi_i^m(T_m) = \frac{y_i^{1/T_m}}{\sum_{i=1}^N y_i^{1/T_m}}, \quad i \in \{1, \dots, N\} \quad (4.9)$$

The function α is a monotonically increasing function of T_m that takes values

CHAPTER 4. STOCHASTIC SEARCH

in the range $[\frac{1}{N}, 1]$ as T_m varies from $[0, \infty]$. The annealing temperature T_m can be obtained by applying the inverse of the function α to a predefined sequence of values α_m . In other words, by finding the value of T_m that would cause $\alpha(T_m)$ to equal α_m .

$$T_m = \alpha^{-1}(\alpha_m). \quad (4.10)$$

Since α is a monotonic function, its inverse is guaranteed to exist. The inverse can be obtained by solving a zero crossing or a local optimization problem. The parameters α_m and ξ_m (regularizer) determine the annealing schedule. We set those parameters by the following power law

$$\alpha_m = \eta \lambda^m, \quad \xi = \text{diag}\{\sigma_1^2, \sigma_2^2, \dots, \sigma_d^2\}, \quad \xi_m = \alpha_m^2 \xi \quad (4.11)$$

Here, each entry of the diagonal matrix ξ is the square of half the maximum change in the corresponding state estimate expected between successive frames. The significant difference between our schedule and that used by APF is that APF sets the parameter α_m to 0.5 for all m . Thus $\eta \in (0, 1)$, $\lambda \in (0, 1)$, N^0 , M and C are the external parameters used by our method. Those parameters are dependent upon the state space dimension d and the density $P(x)$. Figure 4.4 shows the annealing schedule relative to the start temperature for PA and APF.

The inferred MoG is expected to approximate the true density $P(x)$ and is expected to approach a delta as a result of annealing. Though there are methods that can exactly find the mode of a mixture of Gaussian [124], we find that in practice, the distribution in the last layer has a single dominant mode and several insignificant ones. Hence, at the end of the last layer, the state is estimated as the mean vector μ_c^M of the mixture component with highest weight ω_c^M .

$$\hat{X} = \mu_c^M, \quad c = \arg \max_{i \in \{1, \dots, C\}} \omega_i^M \quad (4.12)$$

4.3. PARAMETRIC ANNEALING

Figure 4.4: Relative annealing temperature used by PA (in red) and APF (blue) is plotted against the annealing layer. It can be observed that PA performs a slow annealing as evident from the initial slope as well as a deep annealing as observed by the tail.

4.3.1 Synthetic example

To provide insight into PA, we demonstrate its use on the synthetic example shown in Figure 4.5. A univariate multi-modal density obtained by mixing three Gaussians with mean $(-3, 0, 3)$, variance $(1, 0.5, 1)$ and weights $(0.5, 0.1, 0.4)$ is used. Starting with an initial estimate of 0, PA is used to search for the optimum with 10 layers (M) and 3 mixture components (C). Figure 4.5 shows the samples with state value x_i plotted against weights π_i^m , along with the inferred parametric model. It is observed that the inferred density becomes multi-modal in layer 5. Also, towards the last layer, the shape of the delta function emerges in the global optimum and it is so close to the annealed original distribution (magenta curve) that the inferred density (green curve) hides it. It may be seen that the state estimate converges to the true value towards the last layer.

CHAPTER 4. STOCHASTIC SEARCH

Figure 4.5: A demonstration of parametric annealing. The figure shows in red the samples x_i plotted against the value of π_i^m in different layers. The true density $P(x)$ is shown in blue, the effect of annealing $P(x)$ is shown in magenta, and the inferred density $q(x; \theta)$ is shown in green, all up to a scale.

4.3. PARAMETRIC ANNEALING

Figure 4.6: Scaling properties of the PA (blue) in comparison to the APF 1000 (green) and APF 500 (red). The plots shows the mean (top row) and deviation (bottom row) of improvement in log likelihood for different values of dimension d , number of components k and range parameter s respectively. It can be observed that our method improves the likelihood more than APF, and that the deviation in improvement is almost the same.

4.3.2 Scaling properties

In this section we analyze how our method scales with the dimensionality, multi-modality and the range of search by comparing it with APF on a simple scalable problem. The problem used for the comparison is to let the two stochastic procedure search for the optimum state in a MoG. Inspired by [125], we used a generative method to model the MoG used as the search objective. We assume the search objective θ (which is a MoG) to be a random draw from a distribution $P(\Theta)$. The distribution $P(\Theta)$ is sampled using a generative method i.e., the mixture weights are drawn from a Dirichlet distribution of uniform prior, the inverse covariances are drawn from a Wishart distribution with d dimensional identity as scale matrix and the mean vectors are drawn from a d dimensional Gaussian with a scaled identity matrix (by scale s) as covariance and zero mean.

The variable d , the number of mixture components k and the scale of the covariance s control the dimensionality, multimodality and search range of the objective. We made both the procedures start from the origin of

CHAPTER 4. STOCHASTIC SEARCH

the Euclidean space $\mathbb{R}^d$ and search for the optimum. We considered the difference in the log likelihood between the end state and the start state to be a random variable I (for improvement in the log likelihood) that acts as an indicator of how well the procedure performs. Since the procedures are stochastic in nature we ran the procedure several times on different samples from $P(\Theta)$ and considered the mean and standard deviation of I to be indicative of the performance.

Figure 4.6 plots the scaling properties of the proposed method in comparison to the APF. We compared our method to two configurations of APF. The configurations are APF 1000 ($M = 5, N = 200$) and APF 500 ($M = 5, N = 100$), where M and N are annealing layers and particles per layer respectively. Our method was configured with 438 samples ($438 = N^0 + M * C = 150 + 24 * 12$). The mean and standard deviation of the random variable I is plotted against various values of dimension d , components k and scale factor s . Higher mean value of I and lower deviation are better. The result might seem to show that all procedures somehow perform better in higher dimensions and as the range increases, but this is not true. It is caused by the vast change in the log likelihood in higher dimensional Gaussian mixtures and over long search ranges. However, it can be observed that, in all three tests, our method improves the log likelihood better than APF despite using less than 50% of the samples. It either performs as good as or slightly better than APF with regards to the deviation of improvement.

4.3.3 Sensitivity analysis

Both our method, as well as APF, have quite a number of parameters. A standard set of parameters proposed in [26] is used in most studies involving APF [15]. However, how these parameters affect the tracking performance has not been studied. This motivated us to perform sensitivity analysis on the parameters of our own algorithm since it would provide insight into how those parameters affect tracking. We performed this study in the same objective as the previous experiment. The test was performed by varying each parameter η , λ , N^0 , M and C around a small range from the default parameter used in previous test and observing the mean and deviation of the variable I . Figure 4.7 shows the results of sensitivity analysis. It can

4.3. PARAMETRIC ANNEALING

Figure 4.7: The plot shows the mean improvement and deviation for various settings of the parameters of our procedure. The parameters N^0 , M and C refer to initial number of samples, number of annealing layers and the number of new samples introduced per layer respectively. The parameters η and λ define the power law used to control the rate of annealing.

CHAPTER 4. STOCHASTIC SEARCH

be observed that our algorithm is largely insensitive to the initial number of samples N^0 . In fact, it slightly improves as the initial number of samples reduce, which may be understood by analyzing Figure 4.2. When the initial number of samples are higher, the search is ineffective since the samples hold back the search procedure until they are annealed to a suitable temperature.

An increase in the number of annealing layers M improves the performance, this justifies our choice of slow annealing. However it can be observed the deviation shows a slight decreasing trend as the number of layers reduce. We believe this is due to the cumulative effect of the randomness introduced in every layer. Hence a suitable trade-off between the two is necessary. An increase in the number of new samples C introduced in a layer improves the performance, this is expected since when there are more samples the search should be better. A decrease in the parameters η and λ that define the annealing schedule shows an improvement in performance, indicating that the deeper the annealing, the better the performance of the search procedure. This is expected since, as shown in Section 4.2, a deep annealing schedule is necessary to ensure that the new samples search the state space effectively.

4.4 Tracking framework

In this section, we describe the human pose tracking framework that is used both for evaluating PA, and also for the study on parameter selection described in Section 4.6.

We use the twist representation [29] to model individual joints in our kinematic model. The individual rigid objects that form the kinematic model are connected by a kinematic tree with the torso at its root, as commonly done in recent work [14]. The rigid objects are made of tapered cylinders with elliptical cross sections [15]. The articulated model is assembled in world coordinates by the ordered recursive application of the transformation matrices corresponding to the twists in the kinematic tree.

In order to compare with APF and ISA, we use the silhouette-based cost function χ described in [15] in this work. Let Syn_c^x represent the projection of the kinematic model parameterized by x to camera c , and Sil_c represent the silhouette obtained after background subtraction from camera c , then χ

4.5. APPLICATION TO HUMAN POSE TRACKING

(negative log likelihood) is expressed formally as

$$\chi(x) = \sum_{c=1}^C \frac{1}{|Syn_c^x|} \sum_{i \in Syn_c^x} (1 - Sil_c(i)) + \frac{1}{|Sil_c|} \sum_{i \in Sil_c} (1 - Syn_c^x(i)) \quad (4.13)$$

We used a median-filter based adaptive background subtraction algorithm described in [126] to obtain the silhouettes. The initial reference image is obtained by sub-sampling each validation video sequence to two frames per second and obtaining the median of each pixel from this reduced set of frames. The silhouettes obtained by background subtraction were smoothed by a Gaussian kernel of $\sigma = \sqrt{2}$ to produce a smooth objective [27]. We use silhouettes from 3 RGB cameras and 2 gray scale cameras for tracking.

For the motion prior, we use a constant position model to obtain the prior samples as in [26, 15]. However, the techniques we discuss are applicable with sophisticated prediction methods such as Gaussian process auto-regression models [14, 93] or other simple methods such as constant velocity model [25].

The error measure we use to evaluate results is similar to the one proposed in [15]. Specifically, we place V virtual markers in the model being tracked and on the ground truth obtained from the database. If X represents the list of marker positions of the ground truth as 3D vectors and $\hat{X}$ represent that of the tracked estimate, the error measure can be formally expressed as below

$$D(X, \hat{X}) = \frac{1}{V} \sum_{i=1}^V \|x_i - \hat{x}_i\| \quad (4.14)$$

4.5 Application to human pose tracking

We implemented APF, ISA and parametric annealing (PA) in MATLAB and optimized core computational blocks in C. Most of our code optimization is in the rendering part which is shared between the three methods. We made no special optimization for the inference part of our algorithm except for ensuring that the MATLAB code uses vectorized operations rather than loops. ISA is implemented as described in [14].

CHAPTER 4. STOCHASTIC SEARCH

Configuration	Parameters	Samples
APF^{vh}	$N = 250, M = 15, \alpha = 0.5, \gamma = 0.2$	3750
APF^h	$N = 200, M = 5, \alpha = 0.5, \gamma = 0.2$	1000
APF^l	$N = 100, M = 5, \alpha = 0.5, \gamma = 0.2$	500
ISA^{vh}	$N = 250, M = 15, b = 0.7, \gamma = 0.4$	3750
ISA^h	$N = 200, M = 5, b = 0.7, \gamma = 0.4$	1000
ISA^l	$N = 100, M = 5, b = 0.7, \gamma = 0.4$	500
PA^{vh}	$N^0 = 150, M = 85, C = 17, \eta = 0.74, \lambda = 0.95$	1595
PA^h	$N^0 = 150, M = 50, C = 12, \eta = 0.6, \lambda = 0.9$	750
PA^l	$N^0 = 150, M = 24, C = 12, \eta = 0.6, \lambda = 0.9$	438

Table 4.1: The nine algorithm configurations used for the test are shown.

The S1, S2 and S3 jogging sequences from the Human Eva I dataset are used for comparison. We obtain the tracking results from three configurations of each algorithm, referred to as very high (vh), high (h) and low (l) configuration. The configuration parameters are listed in Table 4.1. The APF^h configuration parameters are the ones used in [15]. However, we modified the adaptive diffusion kernel parameter γ to 0.2 since it was found to make a significant impact in a recent study [14]. We use the same number of layers and samples per layer as APF for ISA, since it allows the two methods to be compared. Other parameters of ISA are as described in [14]. Figure 4.8 shows the tracking results for the configurations described in Table 4.1. Our results using APF on Human Eva I are comparable to those reported in [15]. Overall statistics for the tests on all three sequences are provided in Table 4.2.

We also show in Table 4.2 a variant of our method, referred to as PAK, which uses a kernel density rather than adaptive diffusion to generate new samples. The purpose of PAK is show the effectiveness of the adaptive diffusion. The table, in addition to the time averaged means and deviations for each sequence and configuration, reports the overall (sequence averaged) mean error and deviation. The results show that PA has lower overall error in each configuration than the other algorithms. The overall trend that we observe is consistent with those reported in [14]. We observe that ISA, which is a more recent technique, performs better than APF, and our method performs better than ISA. We also observe that the configuration PAK does not perform as well as PA, indicating that adaptive diffusion results in a significant improvement.

4.5. APPLICATION TO HUMAN POSE TRACKING

Figure 4.8: The figure plots the tracking error (mm) of each algorithm against the frame number for the S2 jogging sequence (in color). Low configurations is shown in (a), high configurations in (b), and very high configuration in (c). Mean tracking error for 8 runs of each procedure is shown, and shows that our method has the lowest error on this sequence in each configuration. Clearly, increasing the number of samples reduces the error.

Figure 4.9 shows the tracked model for six of the configurations, superimposed on frame 241 of camera 1 for S1 jogging sequence. More frames from the same sequence are presented in [17]. It can be observed from the figure that the legs are tracked quite accurately. However some error can be observed in the arm pose for all configurations. We believe this is due to lack of constraints in the cost function for tracking arm. It can be noticed that, excluding the error in arm pose, configurations that use higher number of samples have a lower error. However, for the proposed method, both the configurations have relatively lower error.

CHAPTER 4. STOCHASTIC SEARCH

		Time averaged mean tracking error $\pm \sigma$					
Configuration		S1 Jog (mm)	S2 Jog (mm)	S3 Jog (mm)	Overall (mm)	Percent of APF	Runtime (sec)
Low	APF^l	151 $\pm$ 10	139 $\pm$ 22	156 $\pm$ 27	148 $\pm$ 20	100%	16.3
	ISA^l	128 $\pm$ 13	103 $\pm$ 15	110 $\pm$ 12	113 $\pm$ 14	76.4%	16.3
	PA^l	131 $\pm$ 17	98 $\pm$ 4	109 $\pm$ 10	112 $\pm$ 10	75.7%	14.9
	PAK^l	167 $\pm$ 22	137 $\pm$ 38	155 $\pm$ 37	153 $\pm$ 32	103%	14.9
High	APF^h	147 $\pm$ 14	120 $\pm$ 14	136 $\pm$ 23	134 $\pm$ 17	100%	32.7
	ISA^h	126 $\pm$ 17	99 $\pm$ 14	106 $\pm$ 12	110 $\pm$ 14	82%	32.7
	PA^h	121 $\pm$ 17	95 $\pm$ 5	104 $\pm$ 15	106 $\pm$ 12	79.1%	26.3
	PAK^l	154 $\pm$ 31	130 $\pm$ 19	119 $\pm$ 18	134 $\pm$ 22	100%	26.3

Table 4.2: Overall statistics of the six configurations and the three sequences are shown in the table. Best results in low and high configurations are shown in bold. Overall (average over sequence) mean error and standard deviation is shown in absolute value and percent relative to APF for the two configurations. It can be noticed that PA has the least overall error.

Figure 4.9: Tracked model superimposed on frame 241 of camera 1 for S1 jogging sequence. The model fit can be observed to be worse for low configurations in comparison to high configurations. It can be seen that both the high and the low configuration for the proposed method has a relatively lower error, in comparison to other algorithms.

4.6 Parameter Selection

In last section, we demonstrated our method on a human pose tracking dataset and showed that it performs better than the existing techniques in the literature. However, one concern a user might have is the need to choose the right set of parameters. Indeed, the search performance of all stochastic search procedures (including our own) is highly dependent upon the parameters used for tracking [14]. There is no framework in the current literature to estimate the parameters optimally, though it is recognized as an open problem [26]. This motivated us to devise a systematic framework that would enable the optimal selection of tracking parameters when provided with a training video sequence. The framework we propose is applicable to our method (PA), as well as to other stochastic search procedures such as APF [26] and ISA [92]. Due to the stochastic nature of the search, our framework assumes that the articulated motion and the tracking performance are random variables that interact with the parameters of the tracking algorithm. The parameters are treated as non-random unknown variables that need to be estimated. We choose the parameters such that they track the expected articulated motion accurately. Our framework is described in detail in this section.

4.6.1 Overview

We consider tracking to be a global optimization problem in a state space $X \in \mathbb{R}^d$. Let $\chi_t : \mathbb{R}^d \rightarrow \mathbb{R}$ be a cost function to be minimized by the tracking algorithm at time t . Formally, the tracking problem at a time instance can be expressed as,

$$\hat{X} = \arg \min_X \chi_t(X) \quad (4.15)$$

It is well known that χ_t is a non-convex, high dimensional function that generates a significant computational overhead making the tracking problem hard [26, 15]. In a real tracking scenario, the optimization is dynamic and there is continuity in state evolution resulting in a trajectory in the state-space. For the purpose of parameter selection, we ignore the continuity and assume that the cost function χ_t is a random function drawn at each time instance from an underlying distribution Q in function space.

CHAPTER 4. STOCHASTIC SEARCH

The distribution Q captures the randomness introduced by aspects such as unpredictability in human motion, lighting variation, image sensor noise and residual error introduced in previous time step. Each sample drawn from the distribution Q is a high dimensional non-convex function in the d dimensional state space and a random start estimate from which global optimization is started. In other words, we consider χ_t at various time instances referred in short as χ to be independent and identically distributed samples drawn from the distribution Q .

Let Err be a random variable that acts as an indicator of the tracking performance. It measures the random error introduced by the tracking algorithm in the current time step. Let $\mathcal{A} \in \{\mathcal{A}_j, j \in 1, \dots, A\}$ be the tracking algorithms under consideration and $\mathcal{S}_j$ be the parameters of the corresponding algorithm. We treat the parameters of the algorithm as unknown non-random variables. In a real tracking scenario, the variables $\chi \sim Q$, Err and $\mathcal{S}_j$ interact, making selection of parameters very difficult. Hence we marginalize over χ to obtain the distribution of Err parameterized by $\mathcal{S}_j$. This enables one to measure the marginal error introduced by the tracking algorithm as a function of the parameters of the algorithm. Formally

$$P_{\mathcal{A}_j}(Err; \mathcal{S}_j) = \int P_{\mathcal{A}_j}(\chi, Err; \mathcal{S}_j) d\chi \quad (4.16)$$

The subscript $\mathcal{A}_j$ indicates that the distribution is specific to the algorithm $\mathcal{A}_j$. The marginalization can be done non-parametrically by generating samples from Q (Section 4.6.2). Let $Ovh_{\mathcal{A}_j}$ be a deterministic scalar valued function of $\mathcal{S}_j$ that indicates the computational cost for the given parameter setting. Using the distribution $P_{\mathcal{A}_j}(Err; \mathcal{S}_j)$ and the function $Ovh_{\mathcal{A}_j}$, we define a cost function as below

$$C_{\mathcal{A}_j}(\mathcal{S}_j) = W^T \nu \quad (4.17)$$

$$W = [Ovh, V(Err), M(Err)]^T \quad (4.18)$$

Where $\nu \in \mathbb{R}^3$ is a vector that decides the relative importance of the individual terms that make up the cost function, $V(Err)$ and $M(Err)$ are statistics extracted from $P_{\mathcal{A}_j}(Err; \mathcal{S}_j)$. In other words the cost function is a conic combination of three terms, namely $Ovh_{\mathcal{A}_j}$, variance $V(Err)$ and the mean $M(Err)$ of the samples above 75th percentile of the random variable

4.6. PARAMETER SELECTION

Figure 4.10: The histogram shows the distribution of weighted Euclidean distance between poses of consecutive frames. It can be seen that the pose change is in a typical range, but the tail of the distribution is often hard to track.

Err. We found that minimizing the mean of the worst performing samples, rather than the mean of the entire distribution, results in better generalization. In essence, we are applying a minimax principal that minimizes the maximum error. Thus we pose parameter estimation problem as that of an optimization problem that aims to minimize the cost $C_{\mathcal{A}_j}$ as a function of $\mathcal{S}_j$. Formally, the optimal parameter to track a sequence $\hat{\mathcal{S}}_j$ is obtained as below

$$\hat{\mathcal{S}}_j = \arg \min_{\mathcal{S}_j} C_{\mathcal{A}_j}(\mathcal{S}_j) \quad (4.19)$$

Though not explicit, the parameter estimated for an algorithm $\mathcal{A}_j$ is a function of the articulated motion and the frame rate of the video sequence. They interact via the sampling process that generates χ for marginalization in eq. (4.16).

4.6.2 Random variables and Sampling

The variable χ is assumed to be the cost function presented to the tracking algorithm at a time instance. More specifically, χ is the negative log likelihood (NLL) described in Section 4.4. We assume a training video sequence, along with the ground truth, is available to estimate the parameters. Hence

CHAPTER 4. STOCHASTIC SEARCH

the most naive way to generate samples from Q would be to pick a frame at random, use the ground truth for the frame as starting location, and the cost function corresponding to the next frame as the sample χ . However we found that such a scheme does not generalize well to the real tracking problem.

The behavior could be explained by analyzing the histogram of weighted Euclidean distances (weights inversely proportional to σ_i used in eq. (4.11)) between poses of consecutive frames in a training video sequence. Figure 4.10 shows such a histogram. It can be observed that majority of the state transitions are in a typical range, while there are few outliers in the distribution's tail which have a large change in pose. When samples are generated at random, uniformly through the entire sequence, the search performance of the tracking algorithm on the outliers do not have a large impact on the overall cost $C_{\mathcal{A}_j}$. Consequently, the tracking algorithm starts accumulating more error when such outliers are presented in a real tracking scenario, and the training fails to generalize. Hence, our sampling method generates samples such that the weighted Euclidean distances between the states fall in the top 25th percentile.

The variable Err acts as an indicator of the performance of the tracking algorithm. We define Err to be the difference in cost χ between the estimate and the global minimizer. Formally

$$Err = \chi(\hat{X}) - \min_X \chi(X) \quad (4.20)$$

In our experiments, synthetic silhouettes generated from the ground truth are used for training, hence the global minimum can be obtained exactly.

4.6.3 Optimization

Due to the stochastic nature of the search, the cost function $C_{\mathcal{A}_j}$ is an expensive, noisy, black box objective function. The parameters include integer and real variables. However for simplicity, we treat it as a function on $\mathbb{R}^{|\mathcal{S}_j|}$, where $|\mathcal{S}_j|$ represents the number of parameters used by algorithm $\mathcal{A}_j$. We added bounds to the parameters of each algorithm resulting in a polytope in the space of $\mathcal{S}_j$. In addition to the bounds, a non-linear constraint on

4.7. PARAMETER SELECTION RESULTS

Ovh_{A_j} was encoded in the objective with a hard margin which when violated results in a very large cost.

There are many techniques available in the optimization literature that is suitable to optimize a noisy, expensive function. We experimented with two methods namely, Generalized Pattern Search [127] and a global optimization technique with Gaussian process [128]. The former is based on positive basis theory. The optimization procedure constructs a mesh with a set of positive spanning directions which are then used to select the subsequent iterates of the procedure. The later is an instance of response surface methods that estimates the response surface of the optimization objective in order to reduce the number of evaluations of the expensive function. In our experiments both the methods converged to similar solution with varying overhead.

We compare the three stochastic search algorithms, APF [26], ISA [92] and PA (Section 4.3). The parameters $\mathcal{S}_j$ used for each algorithm are described below.

- $\mathcal{S}_{APF}$ is the number of samples N , number of layers M and the fractions γ and α that control the adaptive kernel and the annealing rate.
- $\mathcal{S}_{ISA}$ is the number of samples N , number of layers M , the constant b that control the monotonic annealing schedule and the fraction γ for the adaptive kernel.
- $\mathcal{S}_{PA}$ is the number of initial samples N^0 , number of new samples in subsequent layers C , number of layers M , the fractions η and λ that control the annealing schedule and the regularizer.

4.7 Parameter Selection Results

In this section we apply our framework to estimate the optimal parameters for various tracking algorithms using data from Human Eva I database. We use two different error measures for comparing the algorithms. Ground Truth Marker Error (GME) is the error measure described in Section 4.4. Though this is the most common error measure, we observe that there is a statistical bias between the ground truth and the minimizer of the cost func-

CHAPTER 4. STOCHASTIC SEARCH

tion. Hence it is suitable for reporting long term tracking trends and less reliable when it comes to discriminating finer details. Negative Log Likelihood (NLL) is the value of the cost function at the converged point. This can discriminate finer tracking error between different search algorithms. However, it assumes that the minimizer of the cost function unambiguously corresponds to the true pose, which in general is not true. More specifically, with the bidirectional silhouette based likelihood used in this study, there are not enough constraints for the arm pose. Consequently, the NLL could be reduced by hiding the arm behind the torso. However, that would result in an incorrect pose in spite of having a lower cost. Hence we report both GME and NLL in this work.

We refer to the framework described in Section 4.6.1, which verifies the parameter setting on a randomly selected set of frames, as the “training” problem. The sequential human pose tracking that is performed as in the current literature [15, 14], i.e. on every frame, is referred to as the “tracking” problem. Each sequence in Human Eva I database has separate data for training and validation. Hence we used the training and validation data from the S2 jogging sequence for the “training” and the “tracking” problems respectively. We used silhouettes synthetically generated from the ground truth for “training”. However, for “tracking” we used silhouettes extracted from the observation using methods described in Section 4.4.

4.7.1 Parameter learning for high tracking accuracy

In this section we demonstrate parameter selection that emphasizes reduction of tracking error. In order to achieve this we chose the mixture proportions ν such that the quantities $M(Err)$, $V(Err)$ and Ovh are given a weight of 3, $1/10$ and $1/1500$ respectively. The weights were chosen heuristically based on the typical range of the quantities in W . We trained ISA and PA with the above mentioned training sequence and applied the optimal parameter setting obtained on the tracking problem. We also include tracking results with the parameter setting used in [14] as APF^d and ISA^d (d indicates they are default configuration results). Table 4.3 shows the statistics obtained during training and tracking for the respective configurations. The configurations obtained by training are appended with a superscript a

4.7. PARAMETER SELECTION RESULTS

Method	Training Statistics				Tracking Statistics		
	$M(Err)$	$V(Err)$	Ovh	C_{A_i}	GME	NLL	Time(sec)
APF ^d	0.45	0.08	3750	1.282	99.0 ± 5.1	29.9 ± 0.3	95.5
ISA ^d	0.87	0.16	3750	4.608	99.2 ± 7.6	29.7 ± 0.3	94.5
ISA ^a	0.84	0.15	2169	1.330	94.2 ± 7.5	30.2 ± 0.4	54.7
PA ^a	0.31	0.07	1595	0.662	92.6 ± 5.1	30.3 ± 0.4	33.5

Table 4.3: Training and tracking statistics for parameter learning that emphasizes high accuracy. Best results are shown in bold. It can be observed that ISA has the lowest NLL when optimized for high accuracy. However, it has a high GME due to ambiguities inherent in χ .

to highlight they are optimal parameters for high accuracy.

It can be observed from Table 4.3 that for ISA our optimization procedure has converged to a state which is Pareto optimal compared to the default parameter setting (ISA^d). Moreover it has 40% lower computational overhead than the default parameter setting. It can be noticed that for PA, the procedure converged to a state that is Pareto optimal with respect to ISA. In addition, it can be discerned from the table that the training generalizes quite well in terms of the error measure GME, i.e., on the actual tracking problem, the estimated parameters perform comparable to the training problem. However, the default APF configuration performs well on the training problem though it performs somewhat poorly on the tracking problem. We believe this is caused by the fact that the tracking problem has higher multimodality than the training problem.

The error measure NLL shows that the default configurations perform a better optimization than the trained ones, which does not coincide well with the measure GME. On analysis we found that the default configuration searches a wider volume of the state space, consequently it switches between correct and incorrect hypothesis (has a low cost χ but does not correspond to the true pose). This results in the higher GME. The test provides an interesting insight that when the observation is less uncertain, it would be preferable to go for the configurations ISA^a and PA^a. However, when the observation is highly uncertain, configurations such as ISA^d and APF^d are better since they would recover the right hypothesis though they may momentarily jump to incorrect one. The difference between configurations ISA^d and APF^d is less pronounced in this test, however on an average

CHAPTER 4. STOCHASTIC SEARCH

Method	Training Statistics				Tracking Statistics		
	$M(Err)$	$V(Err)$	Ovh	C_{A_i}	GME	NLL	Time(sec)
APF ^o	0.83	0.17	696	1.531	101.4 ± 7.7	30.4 ± 0.4	17.9
ISA ^o	1.35	0.26	749	2.110	98.3 ± 8.9	30.8 ± 0.4	19.1
PA ^o	0.80	0.17	674	1.481	95.2 ± 3.8	30.9 ± 0.3	17.8

Table 4.4: Training and tracking statistics for parameter learning that emphasizes low overhead. Best results are shown in bold. It can be observed that PA performs the best when optimized for low overhead as evident from training and tracking statistics.

ISA performs better than APF as shown in section 4.5 which confirms the study in [14].

4.7.2 Parameter learning for low tracking overhead

Whereas in last section we performed parameter selection with the goal of high tracking accuracy, in this section we apply our method with the goal of low tracking overhead. To this end, we chose the mixture proportions ν such that the quantities $M(Err)$, $V(Err)$ and Ovh are given a weight of 1, $1/10$ and $1/1000$ respectively. We trained APF, ISA and PA on the same sequence as before. The statistics from the training and the tracking problem are presented in Table 4.4. The configurations obtained by training are appended with a superscript o to highlight they are optimal parameters for low overhead. We can observe from the table that our method has converged to a state which is Pareto optimal when compared to ISA and APF.

It can be observed from Table 4.4 that the training generalizes reasonably well. The configuration PA^o has the least error in training as well as tracking. The results from APF and ISA are slightly mixed. We believe this may be due to higher uncertainty in the objective used for tracking. It can be noticed that the tracking results obtained for subject 2 jogging after training for PA and APF is better than the high configuration shown in Table 4.2. Tracked models obtained using configuration PA^o superimposed on video input from camera 1 are shown in Figure 4.11.

4.8. STOCHASTIC SEARCH ON A MANIFOLD

Figure 4.11: Tracked output of Subject 2 jogging sequence using configuration PA^o . It can be noticed that the tracked model is a good fit to the subject in the video.

4.8 Stochastic search on a manifold

In this chapter, we discussed stochastic search procedures for human pose tracking. These methods are general global optimization tools that can be applied to any optimization problem. Until now we applied these procedures to optimization problems defined in Euclidean space. In this section, we present an application of these procedures to a more general non-convex optimization problem in a manifold S^n , $\{x \in \mathbb{R}^{n+1} : \|x\| = 1\}$.

Spherical harmonic coefficients have been used to construct a rotation, translation and scale invariant descriptor for 3D rigid surfaces [130, 131]. Furthermore, the coefficients can be used to identify bilateral plane of symmetry [18]. This is achieved by formulating an optimization problem in the manifold S^3 , which is an isomorphic cover for the group $SO(3)$ that parametrize rotations and reflections in 3 dimensions. The global optimum

CHAPTER 4. STOCHASTIC SEARCH

Figure 4.12: The cost function for the bilateral symmetry plane estimation. The horizontal and vertical axis correspond to α and β values in an Euler angle representation of the group $SO(3)$. The optimal values are color coded as dark red.

Figure 4.13: A set of samples generated from three Von Mises Fischer distributions on $\mathbb{S}^2$ [129]. The samples in blue, green and red have a low, medium and high value of the concentration parameter κ respectively. The mean directions μ are shown by the respectively colored lines.

of the problem correspond to the best plane of bilateral symmetry for the rigid 3D surface. The cost function used for symmetry plane estimation is shown in Figure 4.12.

The stochastic search procedures we discussed in this chapter such as APF, ISA and PA are configured to perform global optimization in an Euclidean space. There are two aspects of these algorithms that depend on the space. They are the kernel that is used for sample generation and the estimation method for the optimal state at the end of the procedure. For Euclidean space a Gaussian kernel is used and the state is estimated using the maximum likelihood estimation procedure for Gaussian mean (which is the sample mean). For the manifold $\mathbb{S}^n$ a suitable choice for the kernel would

4.9. SUMMARY

be the Von Mises Fischer distribution, which can be expressed as below

$$f(x; \mu, \kappa) = C_p(\kappa) \exp(\kappa \mu^T x), \quad (4.21)$$

where κ is the concentration parameter, μ is the mean direction and $C_p(\kappa)$ is the normalization constant. A set of samples generated from the distribution on $\mathbb{S}^2$ for three different values of κ and μ is shown in the Figure 4.13. Samples can be generated from the distribution using the method described in [132]. Furthermore, given a set of samples $x_i, i \in \{1, \dots, N\}$ that can be well approximated by the Von Mises Fischer distribution, maximum likelihood estimate [133] for the mean direction μ can be obtained as

$$\mu = \frac{\sum_{i=1}^N x_i}{\|\sum_{i=1}^N x_i\|}. \quad (4.22)$$

Using the Von Mises Fischer kernel and the MLE estimate, the stochastic search procedures such as APF, ISA and PA can be used for global optimization in $\mathbb{S}^n$. In order to perform exhaustive search in the manifold, one could construct an annealing schedule such that the concentration parameter κ is 0 in first layer resulting in a uniform distribution on $\mathbb{S}^n$. The parameter κ should be increased in successive layers in order to enable localization of the global optimum. The average results of the global optimization procedure for various input datasets are presented in Table 4.5. Lower values indicate a more optimal estimate for the symmetry plane. It can be observed that stochastic search obtains a more optimal estimate than the other methods. Figure 4.14 shows the symmetry plane estimated by stochastic search for various 3 dimensional shapes.

4.9 Summary

In this chapter, we describe a procedure that improves on APF for tracking a human from a video sequence. Using synthetic examples we demonstrate the problems with APF and show that they are overcome by two novel steps. Namely reusing samples with a suitable annealing schedule and by inferring a parametric form from the samples. We present a synthetic example that demonstrates the proposed method. We then compare the proposed method

CHAPTER 4. STOCHASTIC SEARCH

Input	SH-COV	N-GRID	SH-SS
PSB	8.8	5.6	5.3
S10	10.8	6.9	6.5
SEG	9.6	6.1	5.7

Table 4.5: The table presents comparative results for stochastic search on a manifold. Lower values indicate a more optimal estimate for the symmetry plane. Column SH-COV presents the results obtained by an approximate closed form solution, N-GRID presents the results obtained by splitting the manifold into grids and refining the estimated in each grid, SH-SS present the results obtained by stochastic search. The rows refer to the various shape datasets used for the study, namely Princeton Shape Benchmark (PSB), SHREC 2010 Generic 3-D warehouse (S10) and Mesh Segmentation Database (SEG). Best results in **bold**.

Figure 4.14: Symmetry plane estimated by stochastic search for various 3D shapes are shown in green [18]. The values of E represent the minimum obtained at the estimate.

4.9. SUMMARY

to APF in a simple scalable problem and show that our method consistently performs better than APF. This was followed by sensitivity analysis showing that our algorithms' performance is largely insensitive to the parameter settings. Furthermore, we present human pose tracking results using data from the Human Eva I dataset that show the benefit of using our algorithm. We then present a systematic framework that enables parameter selection for human motion capture system. We validate the proposed framework using the data from the Human Eva I dataset. Finally, we present an extension to the stochastic search methods, that enable them to perform global optimization in the manifold $\mathbb{S}^n$. We apply it to the problem of bilateral symmetry plane estimation for a 3D surface.

CHAPTER 4. STOCHASTIC SEARCH

Chapter 5

A Dynamic Likelihood Model

5.1 Introduction

In previous chapter, we discussed an optimization procedure to obtain the maximum likelihood estimate for motion capture. In this chapter, we examine the form of the likelihood. Recent studies [15] indicate the need for research into a suitable likelihood (or cost function), both to reduce the computational burden and also to improve the accuracy of tracking. Most state of the art systems on HMC [26, 51, 14, 15] use silhouettes and optionally an appearance model to reconstruct the human motion. These methods have a high computational overhead resulting from rendering and matching. In this chapter, we address the question of what is a suitable cost function for model-based motion capture.

In object recognition [134] and exemplar-based pose estimation [135], directional chamfer matching (DCM) has been in use. The technique uses gradient orientation in addition to position for defining a distance function between edge fragments (see Figure 5.1). DCM is known to have superior discriminative power for articulated objects [134]. Though there are studies that have applied DCM for single-view articulated hand tracking [136], we do not know of any study that has addressed the question of how well DCM can be adapted to multi-view model-based HMC, or how it compares with the state of the art likelihoods used in the model-based HMC.

Furthermore, the performance of DCM based methods is dependent on

Figure 5.1: Illustration of Directional Chamfer Matching (best seen in color). The set of observation (O) and synthesized (S) edge fragments are shown in (a) and (b) as red and green dots respectively. Mean of the distances represented as line segments (in yellow) is the directional chamfer distance between O and S . It can be observed that the length of the line segments increase when O and S are misaligned. The human model is shown in (c).

key parameters such as the orientation specificity parameter [134], which is often learned from training data. Such a learning scheme cannot be applied to HMC, since, as we show, the parameter is sensitive to the position of the subject with respect to the camera view. Though the parameter would not make a significant difference for single-view hand tracking studies such as [136], where the hand moves relatively little with respect to the camera, it would have a significant impact for the state of the art HMC datasets such as Human Eva I and II, where the subjects move significantly relative to the cameras. Furthermore, tracking in a multi-view scenario presents unique opportunities that can be exploited by adapting the standard DCM technique to model-based motion capture. These aspects of multi-view model-based motion capture technique are addressed in this chapter.

Our main contributions are as follows. We formulate a cost function for model-based human motion capture using a dynamic directional chamfer matching (DDCM) scheme. We describe a novel technique that exploits the nature of HMC and the multi-view camera geometry to estimate criti-

5.2. DIRECTIONAL CHAMFER MATCHING

Figure 5.2: Overview of the proposed approach. The current approach matches the synthesized model to an observation. The proposed method extracts directional edge fragments from the observation and the model and minimizes a distance measure between them.

cal parameters of the DDCM. We perform an extensive comparison of the DDCM based cost function with the alternatives used in recent work on HMC using video sequences from the Human Eva I dataset. Our results show that the DDCM based tracker improves the accuracy of the HMC systems significantly. Furthermore, we demonstrate that KD trees may be used for an efficient implementation of DDCM, thereby significantly reducing the computational overhead.

5.2 Directional Chamfer Matching

5.2.1 Overview

Directional chamfer matching is a technique used in object recognition for matching edge fragments [134, 135]. Let $\mathcal{P}$ be the space $\mathbb{R}^2 \times [0, \pi)$ that denotes the centroid of an edge fragment along with the edge orientation. Let $O = \{o_i \in \mathcal{P}\}$ and $S = \{s_i \in \mathcal{P}\}$ represent the respective sets of observation and synthesized edge fragments (see Figure 5.1). The directional chamfer distance between the two is defined as

$$d_{DCM}(O, S) = \frac{1}{|O|} \sum_{o_i \in O} \min_{s_i \in S} d_{\mathcal{P}}(o_i, s_i) \quad (5.1)$$

where $d_{\mathcal{P}} : \mathcal{P} \times \mathcal{P} \rightarrow \mathbb{R}$, described in Section 5.2.3, is a distance metric in $\mathcal{P}$, and $|O|$ is the cardinality of the set O . Let W be a warping function

CHAPTER 5. A DYNAMIC LIKELIHOOD MODEL

parametrized by x . Then the optimal parameter $\hat{x}$ that aligns the edge fragments S synthesized from the model and the observation O is obtained by minimizing the distance d_{DCM} as below

$$\hat{x} = \arg \min_x d_{DCM}(O, W(S; x)). \quad (5.2)$$

When matching edge fragments, the warping function W is generally a planar linear warp. We consider the warp W to be a nonlinear map that assembles the articulated object in its pose in $\mathbb{R}^3$ and projects it to calibrated cameras to obtain the synthesized output S (Figure 5.2 provides an overview). We treat x to be the vector of parameters for the articulated object which is high dimensional as discussed earlier.

The function d_{DCM} is smooth and is known to be robust in the presence of noise [134]. Furthermore, by calculating the distance transform of S , the minimum in (5.1) can be computed in constant time, resulting in linear time complexity for d_{DCM} . Some recent approaches [135] even reach sub-linear processing speeds. However, these methods are not suitable for model-based HMC framework, since d_{DCM} is unidirectional, but for motion capture bidirectional matching is required [15, 137], i.e., the cost function would be $d_{DCM}(O, S) + d_{DCM}(S, O)$. Furthermore, the preprocessing cost is largely ignored in these methods. However, this cannot be ignored in the HMC framework where around 1000 or more simulations will be matched to the observation for processing a single frame. Hence we must adapt DCM to be more suitable for motion capture. We describe an improved method of DCM designed for multi-view HMC in the rest of this section.

5.2.2 Kinematic model and Optimization Procedure

Our kinematic model is made of ten links and 25 degrees of freedom including the global position and orientation. Each link is fleshed with a truncated cone with elliptical cross section [15]. The joints were parametrized by twist vectors [29, 45] that abstract revolute and prismatic (global translation) joints in a single parametric form. Individual links were connected by a kinematic tree with the torso at its root. Homogeneous transformations for individual links were obtained by the ordered multiplication of the exponential maps of the twist vectors $\exp(\theta \hat{\xi})$, corresponding to the ancestors of the

5.2. DIRECTIONAL CHAMFER MATCHING

link in the kinematic tree.

We used the search method defined in [14] for optimization in this work, in order to evaluate the proposed likelihood using a well established optimization technique. However, other methods such as APF, PA, SMC based trackers and other particle based search techniques can also be used with our method.

5.2.3 Likelihood

As outlined in Section 5.2.1, bidirectional d_{DCM} is treated as the energy to be minimized. Therefore, the negative log likelihood is formally expressed as

$$-\ln P(O; X) = \sum_{c \in C} d_{DCM}(O^c, W^c(X)) + d_{DCM}(W^c(X), O^c) \quad (5.3)$$

where C is an index set for the cameras used. The warp and the observation is superscripted with c to indicate that both are dependent upon the camera c .

Let $A_x \in \mathbb{R}^2$ and $A_a \in [0, \pi)$ be the components of $A \in \mathcal{P}$, similarly for B . Let the measure $d_a(a, b)$ represent the shortest angular distance between $a, b \in [0, \pi)$ modulo π , and $d_e(a, b)$ represent the Euclidean distance between $a, b \in \mathbb{R}^2$. Let $\lambda > 0$ be a scale parameter. Then the distance $d_{\mathcal{P}}$ in Eq. (5.1) is defined as

$$d_{\mathcal{P}}(A, B) = \sqrt{\frac{d_e(A_x, B_x)^2}{\lambda} + d_a(A_a, B_a)^2} \quad (5.4)$$

In the Appendix we show that $d_{\mathcal{P}}(.,.)$ is a metric in $\mathcal{P}$. Furthermore, the function $d_{\mathcal{P}}(.,.)$ is almost everywhere differentiable in $\mathcal{P} \times \mathcal{P}$, which is intuitively reasonable since the metric is composed of an Euclidean and a geodesic component. It can be noted that the measure $d_{\mathcal{P}}$ adds weighted Euclidean and angular distances together in a manner similar to [135, 136]. Furthermore, it is shown in [135] that minimizing a weighted sum of both distances outperforms the oriented chamfer matching (OCM) scheme of [134], which adds orientation distance only to the Euclidean minimizer. The difference between DCM and OCM is illustrated in Figure 5.3.

CHAPTER 5. A DYNAMIC LIKELIHOOD MODEL

Figure 5.3: Comparison of OCM and DCM [135]. Matching cost per oriented edge fragment for (a) Oriented chamfer matching [134]; (b) Directional chamfer matching [135]. DCM jointly minimizes Euclidean and orientation distances whereas in OCM the Euclidean distance is augmented with the orientation distance of the nearest Euclidean point.

5.2.4 Complexity and KD trees

The naive implementation of a DDCM based tracker would have a computational complexity of $\mathcal{O}(|S||O|)$ for likelihood evaluation in a single camera, where $|S|$ and $|O|$ are respectively the number of edge fragments in the synthesized set and the observation. The complexity is dominated by the closest point computation in Eq. (5.1), which can be significantly reduced by using KD trees [136].

KD trees are data structures that enable fast computation of nearest neighbor queries for a set of points in Euclidean space (see Figure 5.4). They extend the idea of binary search tree to higher dimensions. Given a set of points, KD trees are built by partitioning the space such that roughly half of the points lie in one half space. Alternating coordinates of the Euclidean space are divided in successive levels as depicted in Figure 5.4. When there is a query for nearest neighbor, the tree is traversed such that the partition in which the query point lies is visited first. The other partition is visited only if the shortest distance from the query point to the other partition is lesser than the distance of the nearest neighbor found so far. This results in efficient computation of nearest neighbors, since only the relevant partitions are visited for random input.

The KD tree technique we just described is suitable only for Euclidean space. However for the manifold $\mathcal{P}$, the joint angle distance has to be com-

5.2. DIRECTIONAL CHAMFER MATCHING

Figure 5.4: An illustration of KD trees [138]. The space partition induced by the construction of the KD tree is shown in (a). The KD tree is represented as a binary tree that splits the space along alternating dimensions in (b).

puted modulo π , and KD trees cannot be directly applied. One can overcome this by augmenting the data with a set of points that are rotated, i.e., for each oriented edge fragment a new one is included with same position but rotated orientation ($a - \pi$, where a is the orientation) [136]. With this change, one could treat the set of oriented edge fragments as if they lie in an Euclidean space, and apply the standard KD tree technique for nearest neighbor lookup.

Building a KD tree with N edge fragments can be done in $\mathcal{O}(N \log N)$ and a nearest neighbor query can be answered in $\mathcal{O}(\log N)$ time for random input. Hence, using KD trees, the complexity of the tracker can be reduced to $\mathcal{O}(|S| \log |S| + |S| \log |O| + |O| \log |S|)$. Building the KD tree for the observation edge fragments can be done as part of preprocessing.

5.2.5 Dynamic parameter estimation

Multi-view model-based HMC systems use multiple cameras for reconstructing the motion of a human subject, with the cameras placed to surround the subject so that occlusion is minimized. The relative importance of the Euclidean distance and the orientation in Eq. (5.4) will significantly change depending on how far the subject is from the camera. This is observed in

CHAPTER 5. A DYNAMIC LIKELIHOOD MODEL

Figure 5.5: Distribution of d_e^2 and d_a^2 when the model is at two different depths. It can be observed that the distribution of d_e^2 moves to the left when the depth of the object is increased, whereas the distribution of d_a^2 is nearly the same (in color).

Figure 5.5 which shows the distribution of the squared Euclidean distance and the scaled orientation distance at the same time instance in different camera views. It can be observed that there is a significant change in the distribution of squared Euclidean distance, whereas the distribution of the orientation distance does not change significantly. An intuitive reason for this is that unlike image domain distances that do change with depth, angles do not change with depth. As a result, if a constant value of λ is used in Eq. (5.4) for tracking, the Euclidean and the orientation components would have varying influence at various time instances and camera views. Hence the parameter λ for each camera must be dynamically estimated based on the distance of the model from the camera. We achieve this by expressing λ as the ratio of the expected value of d_e^2 and d_a^2 , i.e.,

$$\lambda = \frac{\mathbb{E}[d_e^2]}{\mathbb{E}[d_a^2]}. \quad (5.5)$$

We now obtain an analytical expression for λ .

Let us consider a 3D point (X_1, Y_1, Z_1) on the surface of the model and the respective closest point (X_2, Y_2, Z_2) on the surface of the subject in camera coordinate system. Assuming perspective projection for the camera with scale parameters F_x , F_y and camera center C_x , C_y , the projected point

5.2. DIRECTIONAL CHAMFER MATCHING

(x_1, y_1) in the image plane can be expressed as below

$$x_1 = \frac{X_1 F_x}{Z_1} + C_x \quad y_1 = \frac{Y_1 F_y}{Z_1} + C_y \quad (5.6)$$

Similarly x_2, y_2 are obtained. Since the nearest edge fragment pairs in the image domain are approximately at the same depth, we can assume that the two points in 3D space are at the same depth, i.e, $Z_1 = Z_2 = Z$. The squared Euclidean norm of the distance between the two projected points in the image domain can be expressed as below

$$d_e^2 = \left(\frac{X_1 - X_2}{Z} \right)^2 F_x^2 + \left(\frac{Y_1 - Y_2}{Z} \right)^2 F_y^2 \quad (5.7)$$

Furthermore, let us assume the distance between the two points in 3D to be D . Then the above expression can be expressed in polar coordinates as below

$$d_e^2 = \frac{D^2}{Z^2} (F_x^2 \cos^2 \theta + F_y^2 \sin^2 \theta) \quad (5.8)$$

Further, assuming an uniform prior over θ in the range of $[0, 2\pi)$, the expected squared Euclidean distance in the image plane can be obtained as below

$$\mathbb{E}[d_e^2] = \frac{D^2}{2\pi Z^2} \int_0^{2\pi} F_x^2 \cos^2 \theta + F_y^2 \sin^2 \theta \, d\theta \quad (5.9)$$

Solving the integral in Eq. (5.9) and substituting it in Eq. (5.5), λ can be expressed as

$$\lambda = \frac{D^2 (F_x^2 + F_y^2)}{2Z^2 \mathbb{E}[d_a^2]} \quad (5.10)$$

Further, assuming an uniform prior over d_a in the range of $[0, \pi/2)$, we have

$$\mathbb{E}[d_a^2] = \frac{\pi^2}{48}. \quad (5.11)$$

Since $\mathbb{E}[d_a^2]$ is a constant, it can be absorbed into D , resulting in the below expression for λ .

$$\lambda = \frac{D^2 (F_x^2 + F_y^2)}{2Z^2} \quad (5.12)$$

The above equation expresses λ as a function of the depth of the subject Z , camera parameters F_x, F_y and a free parameter D . The parameter D , represents the mismatch between the model used to track and the subject. Since typically the model represents the human body rather than the clothed

CHAPTER 5. A DYNAMIC LIKELIHOOD MODEL

subject, it should be high when the subject wears loose fitting clothes.

An analysis of Eq. (5.12) reveals time and camera dependent aspects embedded in it. The variable Z is indeed a function of time and the camera since it is the depth of the subject at a particular time instance in camera coordinates. The variables F_x and F_y are camera dependent and do not vary with time. The variable D is independent of camera and time. We refer to the DCM scheme that uses λ in Eq. (5.4) estimated by Eq. (5.12) as dynamic directional chamfer matching (DDCM). It is dynamic because the weighting between the Euclidean and the angular distances is continually adjusted by the distance of the subject from the camera. Hence λ is a function of time, camera, and distance.

5.3 Experiments and Results

In this section, we provide the implementation details of our method. Later, we study the effect of various parameters in the cost function and compare the proposed method with the alternative methods used for HMC in recent publications [14, 15, 95]. We use the Human Eva I dataset that provides video and motion capture data for validating the proposed method. Input from 3 RGB cameras and 2 grayscale cameras was used in all the sequences except the Subject 3 jogging and walking sequence. An additional camera input was used for these two sequences, since the segmented video had a blind spot near a corner, where nearly 60% of the foreground was invisible, resulting in a tracking failure. We performed initialization using the ground truth [15]. In order to measure tracking error, we registered a set of markers provided by the ground truth to the model in the first frame. Mean Euclidean distance (in mm) between the registered markers and the ground truth in subsequent frames are reported as the tracking error. We used a constant position model [26] for motion prior.

5.3.1 Observation Edge Fragments and Model Synthesis

Observation edge fragments were extracted from the input images in a pre-processing stage. Figure 5.6 shows the edge fragment extraction pipeline from the observed images. Using the background statistics, we extract the

5.3. EXPERIMENTS AND RESULTS

Figure 5.6: Steps involved in edge fragment extraction from the observed image (in color). Edge fragments from the silhouette and the intensity image are fused to result in the set of observation edge fragments O .

silhouette of the subject from the input image. Furthermore, we extract a set of edge fragments from the edges in the silhouette. An edge fragment has a position vector $\in \mathbb{R}^2$ and an orientation mapped to $[0, \pi)$. Figure 5.6a shows the set of edge fragments extracted from the silhouette. It can be observed that the edge fragments include considerable noise introduced by background subtraction.

In addition to the edge fragments from the silhouette, we extract a set of edge fragments from the observed image within the silhouette region. We removed the edges which are very close to an edge fragment from the silhouette by thresholding the shortest distance given by (5.4). The thresholding ensures that there are not too many edge fragments near the boundary. Figure 5.6b shows the set of edge fragments extracted from the observed image. We used the Canny edge detector for all edge extraction. The edges extracted from the image introduce some noise, but they also provide additional cues such as depth discontinuity which makes tracking robust. Moreover, using orientation cues in addition to the location makes chamfer matching robust to such noise.

We merge the edge fragments from the two input source to obtain the observation edge fragments shown in Figure 5.6c. We prune the merged edge fragments using a fixed threshold on the shortest distance (Eq. (5.4)) to the synthesized output from the prior model. It can be observed in Figure 5.6c

CHAPTER 5. A DYNAMIC LIKELIHOOD MODEL

Figure 5.7: Steps involved in model synthesis. First triangular faces are generated from the model, followed by the extraction of directional edge fragments. In a final step occluded edge fragments are removed.

that there is less noise close to the shoe of the subject when compared to Figure 5.6a as a result of pruning. In our experiments on an average there were 600 observation edge fragments per image frame.

Figure 5.7 shows the steps involved in model synthesis. We perform model synthesis by projecting each truncated cone to the calibrated cameras to obtain quadrilaterals for each body part. The quadrilaterals, which are each divided into two triangular faces, are shown in Figure 5.7a. We retain the x , y and z coordinates for each face in the camera coordinates after projection. Further, we generate a fixed number of equispaced edge fragments on the sides of the quadrilateral. The synthesized edge fragments are shown in Figure 5.7b. We remove all the edge fragments that are occluded by a triangular face using an object space hidden surface removal algorithm. Figure 5.7c shows the set of synthesized edge fragments that are not occluded. In our experiments, we found that occlusion removal is a crucial step necessary for robust tracking. It can be noticed that the pose is better discernible in Figure 5.7c compared to 5.7b. Orientation for the edge fragments are obtained from the equation of the line. In our experiments, we generate 24 such edge fragments per limb, resulting in fewer than 240 synthesized edge fragments, since several edge fragments are occluded in a typical pose.

5.3. EXPERIMENTS AND RESULTS

Figure 5.8: The λ value estimated for the six cameras using Eq. (5.12) for a value of $D = 200$ is plotted for different frames of the Subject 3 walking sequence. It can be observed that the value of λ changes periodically as a result of the walking motion of the subject. The fixed value of $\lambda = 450$, which is approximately the average over time and camera for all the λ values, is plotted as a straight line for comparison (best seen in color).

5.3.2 Effect of parameter λ

In this section, we evaluate the benefit of adaptive λ using Eq. (5.12) on the tracking results. The λ value estimated dynamically is nearly the same as the static value when the subject is stationary near the centroid of the cameras. Therefore, we use the walking and jogging sequences from the three subjects for the evaluation. The evaluation is done by obtaining tracking results with dynamic estimation of λ using the Eq. (5.12) followed by obtaining the tracking results using a static values of λ . We found a value of $D = 200\text{mm}$ to produce good results for dynamic estimation. This exact value of D is not crucial, as we found a range of nearby values to perform similarly. The static value of λ was chosen as the mean of the dynamic values of λ over time and camera, which was found to be near 450. The values of λ estimated dynamically and the static λ are plotted against the frame number for the Subject 3 walking sequence in Figure 5.8. The periodic pattern resulting from the walking motion of the subject can be observed in Figure 5.8. The vantage point of the cameras used in Human Eva I is approximately depicted in Figure 5.9. The symmetry in the λ curves in Figure 5.8 can be understood by noting the symmetric positions of the camera in Figure 5.9.

CHAPTER 5. A DYNAMIC LIKELIHOOD MODEL

Figure 5.9: The approximate Human Eva I placement of the three color cameras (arrows) and three gray scale cameras (triangles) are shown in the figure. The walking area of the subject is depicted by the oval in the middle. The λ values plotted in Figure 5.8 correspond to the cameras depicted in respective colors in the figure. The symmetry in the curves shown in Figure 5.8 can be observed to be caused by the camera position (best seen in color).

Figure 5.10: Tracking error using dynamic estimation of λ and static λ . Mean tracking error and one standard deviation for 5 tracking runs is shown. The results were obtained using a multi-view setup using 6 cameras. It can be observed that dynamic estimation of λ makes the tracking robust.

Overall error mean (mm) $\pm$ std	
$\lambda = 450$	84.35 ± 17.91
Dynamic λ	76.36 ± 5.83

Table 5.1: Overall tracking error using dynamic estimation of λ and static λ . It can be observed that dynamic estimation of λ significantly improves the tracking results.

5.3. EXPERIMENTS AND RESULTS

The tracking results are shown in Figure 5.10 and Table 5.1. It can be observed that a fixed value of λ results in poor tracking performance. However, if λ is dynamically chosen using the Eq. (5.12), the tracking performance significantly improves. For the sequences compared, there is a close to 10% reduction in mean tracking error and 70% reduction in standard deviation as a result of dynamic estimation of λ using Eq. (5.12).

5.3.3 Comparison with other cost function

We compared the proposed DDCM likelihood to two common likelihoods used with model-based motion capture. Bidirectional silhouette (**BiS**) based likelihood was found to produce the best results in [15]. **BiS** for a camera C is composed of two terms,

$$\begin{aligned} -\ln P_{\mathbf{BiS}}(Sil; X, C) = & \frac{1}{|\sum_p Sil(p)|} \sum_p Sil(p)(1 - \Pi(p)) \\ & + \frac{1}{|p \in \Pi(X)|} \sum_{p \in \Pi(X)} (1 - Sil(p)) \end{aligned} \quad (5.13)$$

where Sil is the smoothed silhouette and $\Pi(X)$ is the synthesized projection. The first terms in Eq. (5.13) penalizes the set of pixels in the silhouette that are not explained by the synthesized projection and the second term penalizes the set of pixels in the synthetic projection that are not explained by the silhouette.

Appearance (**App**) based likelihood [14, 95] uses appearance cues in addition to the silhouette (**BiS**). **App** requires the decomposition of the projection $\Pi(X)$ into those of the individual body parts $\Pi_i(X)$, which takes occlusion into consideration. The statistics of individual body parts Ψ_i are extracted from the initial frame and are assumed to remain unchanged. Using the statistics and the decomposed projection, appearance based objective for a camera C is formally expressed as

$$-\ln P_{\mathbf{App}}(I; X, C) = \frac{1}{|parts|} \sum_{i \in parts} D_B(\Psi_i || \Psi(\Pi_i(X), I)) \quad (5.14)$$

where $D_B(X||Y)$ is the Bhattacharya distance between the two distributions, and $\Psi(\Pi_i(X), I)$ is the statistics extracted from the image I in the

CHAPTER 5. A DYNAMIC LIKELIHOOD MODEL

Figure 5.11: Image features used for tracking. It can be observed that there is considerable noise in the image features.

region corresponding to $\Pi_i(X)$. The appearance statistics were represented as histograms. As in [14], we use a and b channels of the CIELab representation of the image, in order to make the likelihood invariant to lighting change. Similar to **DDCM** the overall objective for **App** and **BiS** where obtained by summing the objectives from all the cameras.

Figure 5.11 shows the image features used by the three likelihoods for tracking. The set O of observation edge fragments are shown as red dots. The silhouette extracted from the observed image is shown as the foreground region (non gray valued pixels). The appearance cues are represented by the RGB values of the silhouette region. It can be observed that there is significant noise in the image features used by all the three methods.

Figure 5.12 shows the time and ensemble averaged tracking error and deviation of 5 different runs for each sequence. Our results for **BiS** are comparable to those reported in [28, 12]. It can be noticed that for Subject 1, in general **BiS** has a very high error. We found the reason for this to be the poor model fit (see Figure 5.13) for the subject, which results in a highly multimodal likelihood. Using appearance cues (**App**), improves the tracking performance. Our method (**DDCM**) has a lower tracking error. The tracking performance of Subject 2 is equally good with **BiS** and **App**, since Subject 2 is well segmented from the background. Our method performs significantly better than the other two methods for this subject too. Silhouettes obtained from Subject 3 are highly reliable, but the subject is

5.3. EXPERIMENTS AND RESULTS

Figure 5.12: Tracking error for the twelve sequences compared. The results were obtained using a multi-view setup using 6 cameras. Overall mean error for all the sequences compared is shown in the last group. It can be observed that the likelihood DDCM has the least overall error and it performs better than the other likelihoods in 9 out of 12 sequences.

Figure 5.13: Tracking results for the Subject 1 jogging sequence obtained using DDCM (best result out of five runs).

CHAPTER 5. A DYNAMIC LIKELIHOOD MODEL

Figure 5.14: The (best among five runs) tracked model superimposed on the image for two frames from the Subject 1 and 2 jogging sequences are shown. The three columns correspond to likelihoods **BiS**, **App** and **DDCM** respectively. It can be observed that the pose estimate obtained by our method is more accurate.

5.3. EXPERIMENTS AND RESULTS

	App	BiS	DDCM
Runtime per frame (sec)	29.3	26	8.5
Overall error (mm)			
mean $\pm$ std	96 $\pm$ 14	103 $\pm$ 18	81 $\pm$ 9

Table 5.2: Runtime and overall error for the three likelihoods compared. The proposed method has the lowest error and computational time.

completely dressed in black. As a result there are hardly any edges visible due to depth discontinuity (can be observed in 5.11c). Hence, for this subject our method performs worse for two sequences. It can be noticed that **DDCM** performs better in 9 out of 12 sequences compared. The overall average error (shown in the last group of Figure 5.12 and in Table 5.2) for the proposed method is more than 20% lower than that of **BiS** and 15% lower than **App**. The best tracking result for the Subject 1 jogging sequence, using **DDCM** is shown in Figure 5.13. The best tracking results for selected frames of the Subject 1 and 2 jogging sequences, using the likelihoods **BiS**, **App** and **DDCM** are shown in Figure 5.14.

5.3.4 Computational Overhead

Our HMC framework is currently implemented in Matlab. However, critical computational blocks such as the evaluation of the cost function are implemented in C. We ran the code on a standard PC and the processing time per frame for the three methods compared is shown in Table 5.2. It can be observed that our method is significantly faster than the other two methods.

Since there is no rigorous way to compare the computational overhead resulting from the likelihood evaluation, we provide an intuitive argument. Figure 5.15 shows the data used by the likelihoods such as **BiS**, **App** and **DDCM**. It can be observed that **DDCM** uses significantly less information when compared to **BiS** and **App**. This helps reduce the tracking overhead of the proposed method since both the rendering cost as well as the matching cost is significantly reduced. Further, as we showed in Section 5.2.4, using suitable data structures such as KD trees significantly reduces the complexity and the computation time for matching. Our claim is supported by the fact that the study in [14], despite using a GPU based rendering and native

CHAPTER 5. A DYNAMIC LIKELIHOOD MODEL

Figure 5.15: An intuitive argument for low computational overhead of DDCM. It can be observed that the data used by DDCM is significantly less. Furthermore, the pose expressed by **DDCM** is less ambiguous even for a human observer.

code requires 20 seconds per frame (obtained by interpolating the results reported in [14] for 3500 samples to 1000 samples), which we are able to achieve in 8.5 seconds without using GPU in a mixed Matlab and native environment. Furthermore, despite using less data, **DDCM** is capable of discriminating the pose better. This is observed in Figure 5.15, where the pose expressed by **DDCM** is less ambiguous even for a human subject.

5.4 Summary

In this chapter, we propose an approach to human motion capture using a dynamic directional chamfer matching scheme. We describe a novel technique that exploits the multi-view nature of HMC systems for the estimation of a key parameter for tracking. The dynamic matching is demonstrated to make a significant difference to the performance of the DCM based tracker. We compare the DDCM based likelihood with the alternative methods used in the most recent HMC systems, and show that it reduces the tracking error significantly. We further describe an efficient KD tree based implementation of DDCM that could significantly reduce the computational complexity to matching.

Chapter 6

State Space Decomposition

6.1 Introduction

In Chapter 4 we described a stochastic search technique named PA that improves on the widely used method APF. Both APF and PA are designed to be general optimization methods which are applied to HMC. However, articulated objects such as a human body, have a number of conditional independence properties. For example, one could assume that the head and the leg poses are conditionally independent given the pose of the torso. Existing methods for HMC do not make use of these properties, i.e., they are incapable of extracting a good leg pose from a sample which has poor overall likelihood due to head pose. This is observed in Figure 6.1a. Due to occlusion, these independence assumptions may not hold in a single view. However, most state of the art systems for HMC operate in a multi-view scenario, where these assumptions can be made to improve the tracking performance.

Partitioned sampling [118] is a technique that enables articulated object trackers to decompose the high dimensional state space. It has shown a 50% reduction in the tracking overhead for an articulated hand [118] with fewer (4) degrees of freedom than a human (25-50). However, annealing based methods such as APF [26] have been claimed to perform better than partitioned sampling for high dimensional spaces such as those used in HMC. Though recent studies [15] indicate partitioned sampling to be a promising

CHAPTER 6. STATE SPACE DECOMPOSITION

Figure 6.1: Part a) shows two poor overall poses which have a good fit for specific limbs, and b) shows a new pose extracted from the two poor poses that has an overall good fit.

alternative for HMC, state of the art systems [14, 15] do not make use of such a hierarchical decomposition, due to the lack of a systematic framework.

In this chapter, we present a systematic approach to perform hierarchical decomposition. Though we apply our framework to HMC here, our method is a very general one, and it can be applied to other articulated objects such as a human hand or quadrupeds as well as other optimization problems that can be factorized. Our contributions are as follows. We describe a probabilistic framework to decompose the high dimensional state space of the HMC systems into subspaces of smaller dimension. The decomposition enables partitioned sampling type algorithms for HMC. We extend stochastic search methods (APF [26], ISA [14]) to make use of the decomposition. The idea we present in this Chapter does not reuse samples across layers, which was the main idea behind PA presented in Chapter 4. Though the idea of reusing samples can be fused with hierarchical decomposition, further research is necessary to achieve this. Hence this is left as a future work.

We validate the proposed method using the data from the Human Eva I and II datasets. Our results show that by decomposing the state space, we are able to capture complex human activity more accurately, using only a small fraction of the computational resources as the state-of-the-art systems [50, 3, 14].

6.2. STATE SPACE DECOMPOSITION

The formulation we present in this chapter resemble the ideas pursued in model free methods such as [75]. These methods have used Bayesian networks similar to the one we propose in this chapter. However, they are very different from the one we discuss in this chapter for the following reasons.

- Model free methods cannot ensure anatomically correct reconstruction of the human motion since they rely on soft constraint between limbs opposed to our method which enforces hard constraints.
- In our method, since the conditional independencies are induced by the kinematic model, it has a different set of conditional independencies than model free methods such as [75]. It can be observed by comparing the graphical models.
- The random variables in model-free methods are the 6 degrees of freedom of all the links used in the system resulting in a very high dimensional state space. The random variable in our method is the parameter space of the kinematic model which has a relatively smaller dimension.

6.2 State space decomposition

6.2.1 Overview

Recent model-based methods for HMC [26, 51, 14, 15, 95, 50], approach the problem as a dynamic maximum likelihood estimation of a high dimensional state space $\mathcal{X}$. We make a certain conditional independence assumptions about the likelihood, i.e., we assume the likelihood could be factorized into subspaces $\mathcal{X}_i, i \in \{1, \dots, L\}$ of much smaller dimensions. The factorization is achieved by defining conditional likelihoods for each part rather than an overall likelihood as commonly done in model-based methods [14, 15]. The conditional likelihoods are composed of a model to observation and an observation to model matching cost. However, since the conditional likelihoods are defined for each part, we decompose the observation probabilistically in order to define the conditional likelihoods. Using our conditional likelihoods, pose inference can be achieved using well known algorithms such as

CHAPTER 6. STATE SPACE DECOMPOSITION

Figure 6.2: The graphical model shows the conditional independence assumptions induced by the kinematic model.

non-parametric belief propagation (NBP) [139]. However, this would result in a Bayesian particle based system that approximates the posterior with a set of samples. Since most Bayesian methods such as [118, 27] require a very high number of samples for HMC, we extend the stochastic search methods for inference.

In the rest of this section, we describe our method in detail.

6.2.2 Marginal Likelihood

We achieve decomposition by making a number of conditional independence assumptions. The assumptions are induced by the kinematic model used for tracking. Figure 6.2 shows a directed graphical model [71] representing the conditional independence assumptions made. The decomposed subspaces $\mathcal{X}_i$ are simply the degrees of freedom (DOF) for the rigid objects represented by the nodes in Figure 6.2. In order to illustrate model decomposition, let us consider the example of the lower left arm. After marginalizing the unrelated variables, the likelihood for the lower left arm and its parents is expressed as

$$P(lla, ula, tor) = P(lla|ula, tor)P(ula|tor)P(tor) \quad (6.1)$$

where lla , ula and tor represent the pose parameters corresponding to the lower left arm, upper left arm, and the torso respectively. In order to obtain

6.2. STATE SPACE DECOMPOSITION

the likelihood of lla one can marginalize the above equation as below

$$P(lla = l) = \int P(lla = l, ula = u, tor = t) du dt \quad (6.2)$$

The marginalization can be done numerically by a Monte Carlo approximation of the conditional likelihoods.

6.2.3 Conditional Likelihood

We use Dynamic Directional Chamfer Matching (DDCM) discussed in Chapter 5 in this work. However, the techniques that we discuss can be applied with other image features such as those used in [14, 15, 50]. Let $\mathcal{P}$ be the space $\mathbb{R}^2 \times [0, \pi]$ representing directional edge fragments. Let $O^c = \{o_i \in \mathcal{P}\}$ and $\Pi^c(X) = \{\pi_i \in \mathcal{P}\}$ represent the respective sets of observation and synthesized edge fragments for a specific camera c . The directional chamfer distance between the two for the camera c is defined as

$$\begin{aligned} \psi^c(O^c, X) = & \frac{1}{|O^c|} \sum_{o_i \in O^c} d(o_i, \Pi^c(X)) \\ & + \frac{1}{|\Pi^c(X)|} \sum_{\pi_i \in \Pi^c(X)} d(\pi_i, O^c) \end{aligned} \quad (6.3)$$

where $d : \mathcal{P} \times \mathcal{P}^N \rightarrow \mathbb{R}$, is the distance of the nearest neighbor between an element in $\mathcal{P}$ and a set $\{\mathcal{P}\}$ measured using the metric $d_{\mathcal{P}}$ described in Eq. (5.4), $|O^c|$ and $|\Pi^c(X)|$ are the cardinalities of the sets O^c and $\Pi^c(X)$ respectively. It can be observed that ψ^c is composed of a model to observation and an observation to model matching term. The overall distance measure ψ is defined as the mean of the measure ψ^c from all cameras.

The measure ψ provides a scalar cost that measures the overall match between the observation and the projection. In order to decompose it, we introduce a label l , which is distributed according to the L valued categorical distribution, formally, $l \sim \text{Cat}(L, p)$. The probability of the label indicates the degree of membership of an edge fragment to the individual rigid parts. Let us assume that the probability of the part given an edge fragment x in a camera c , $P(l|x, c)$, is known (described in section 6.2.4). Assuming a uniform prior over the edge fragments given the camera, and using Bayes

CHAPTER 6. STATE SPACE DECOMPOSITION

Figure 6.3: The probability of the edge fragment given the part, $P(x|l, c)$, for head is shown in (a) (high probability in green and low probability in black). The edge fragments for the projection $\Pi(X)$ and the observation O , color coded according to the most probable label assignment $P(l|x, c)$ are shown in (b) and (c) respectively.

rule, one can obtain $P(x|l, c)$.

$$P(x|l, c) = \frac{P(l|x, c)}{\sum_x P(l|x, c)} \quad (6.4)$$

Since the possible values x can take is dependent upon c , here we assume the parameters for c and x are consistent, i.e., the probability $P(x, c)$ is non zero. Figure 6.3a shows the probability of the observed edge fragments for the head. In addition, assuming an uniform prior over the cameras and using Bayes rule, one can obtain the probability of the camera given the part as below

$$P(c|l) = \frac{\sum_x P(l|x, c)}{\sum_{x,c} P(l|x, c)} \quad (6.5)$$

Figure 6.4 shows the probability of the camera for the lower left leg and the upper left arm. It can be observed that the lower left leg shown in cyan is equally visible in all three cameras. Hence the measure $P(c | l = \text{lower left leg})$ is uniformly distributed. Whereas the upper left arm shown in green is nearly occluded in the third and second camera. Hence the measure $P(c | l = \text{upper left arm})$ is small for the third and second cameras.

Using these probabilities the cost for a part l and camera c is expressed

6.2. STATE SPACE DECOMPOSITION

Figure 6.4: The probability of the camera given the part, $P(c|l)$, for the lower left leg and the upper left arm is shown. The edge fragments (in color) which are most likely to be from the upper left arm and the lower left leg are shown in green and cyan respectively.

as

$$\begin{aligned} \phi_l^c(O^c, X) = & E_{P(x=o_i|l,c)}[d(o_i, \Pi^c(X))] \\ & + E_{P(x=\pi_i|l,c)}[d(\pi_i, O^c)] \end{aligned} \quad (6.6)$$

where the first and the second term correspond to the observation to model and the model to observation matching cost respectively. The total cost for the part l is expressed as the expectation of ϕ_l^c with respect to $P(c|l)$. Formally,

$$\phi_l(X) = E_{P(c|l)}[\phi_l^c(O^c, X)] \quad (6.7)$$

Model-based methods in the current literature, take an average over all cameras to obtain the overall likelihood, which is equivalent to considering $P(c|l)$ to be uniform.

The conditional likelihood of a part given its parents is expressed as the cost for the respective part. Intuitively, treating the cost as conditional likelihood makes sense, since the cost for a part is conditioned on the value of the parameters for the parent links in the kinematic model. Formally,

$$-\log P(X_j | \text{parents}(X_j)) = \phi_l(X) + C \quad (6.8)$$

where X_j is a vector of parameters associated with the link j , $\text{parents}(X_j)$ is a vector of parameters for the ancestors of the link j (in the graphical

 CHAPTER 6. STATE SPACE DECOMPOSITION

model), and C is the normalization constant. For example, if j is considered to be the *lla* in Figure 6.2, then the vector X_j would be the parameter for the 1D joint associated with the *lla*, and the vector $parents(X_j)$ would comprise of the parameters for the 9 DOF for the *torso* (6) and the *ula* (3).

The formulation of the conditional likelihood in Eq. (6.8) is only an approximation due to occlusion. However, this is not a bad approximation since we take expectation over multiple cameras. Furthermore, since $P(c|l)$ is not uniform, the cost ϕ_l for the part l is more influenced by the view in which the part is not occluded.

6.2.4 Edge fragment prior

We estimate the prior probabilities for the observation edge fragments using the prior state estimate. A similar decomposition is performed in [50]. However, in our work the observation is a set of directional edge fragments, whereas in [50] it is a silhouette. Let $\bar{X}$ be the prior estimate of the state. The edge fragments corresponding to the different parts can be separated by analyzing the part labels during the synthesis. Let $\Pi_l^c(\bar{X})$ represent a set of synthesized edge fragments corresponding to limb l and camera c . Let $\Pi^c(\bar{X})$ represent the complete set of synthesized edge fragments for camera c . The label probability for the observation edge fragment o_i is formally expressed as

$$\log P(l|o_i, c) = C + \begin{cases} d(o_i, \Pi_l^c(\bar{X})) & \Pi_l^c(\bar{X}) \neq \emptyset \\ -\frac{1}{T} \left\{ \frac{1}{|O^c| |\Pi^c(\bar{X})|} \sum_{\substack{o \in O^c \\ p \in \Pi^c(\bar{X})}} d_{\mathcal{P}}(o, p) \right\} & \Pi_l^c(\bar{X}) = \emptyset \end{cases} \quad (6.9)$$

where T is a constant used to control the uncertainty and C is the normalization constant. The set $\Pi_l^c(\bar{X})$ is empty when the part l is occluded in the camera c . For such an occluded part, we define the probability to be a low nonzero value. Since setting a constant value can make it sensitive to the distance measure being used, we define it to be the mean distance between the set O^c and $\Pi^c(\bar{X})$. The function $d_{\mathcal{P}}$ is a distance metric between directional edge fragments [135], which is typically a convex combination of the Euclidean metric and orientation distance.

6.2. STATE SPACE DECOMPOSITION

The constant T has a significant impact on the edge prior. As T approaches ∞ , the edge fragments become equally likely to be from any part and as T approaches 0, the edge fragments are assigned to a single part with high probability. In general, we observe that reducing T improves the search performance. This is expected since as T approaches 0, the edge prior is more informative. Therefore, the decomposed likelihood is influenced by both the model to observation and the observation to model matching cost. Whereas when T approaches ∞ , as a result of a close to uniform edge prior, the observation to model matching cost in the decomposed likelihood is ineffective. However, when the observation is highly ambiguous, reducing T causes the tracker to get stuck in incorrect hypothesis. Hence T should be chosen as a trade-off between the two extremes.

The label probabilities for the projected model is defined as the Kronecker delta, since the part assignment is known with probability 1. Formally,

$$P(l = j | x = \pi_i, c) = \delta(\pi_i^l - j) \quad (6.10)$$

where π_i^l is the label corresponding to the projected model edge fragment π_i . The respective Figures 6.3b and 6.3c show the most probable label assignment for the synthetic output and the observation for a specific camera.

6.2.5 Inference

We adapted the stochastic search procedures (APF [26], ISA [14]) to make use of the decomposed likelihoods. In this section, we describe the modifications we made. Stochastic search methods start with the predicted estimate of the state for the current frame (obtained using GP regression [14] or simple motion prediction strategies such as constant velocity or position [26]) and construct a sequence of layers. Each layer consists of a set of samples, each of which is a tuple consisting of the pose vector X^j and its respective normalized weight w^j . The index $j \in \{1, \dots, N\}$, where N is the number of samples used in a layer. Sample weights w^j are obtained by evaluating the annealed likelihoods and normalizing as below

$$\log q^j = -\beta^l \psi(X^j), \quad w^j = \frac{q^j}{\sum_{k=1}^N q^k} \quad (6.11)$$

CHAPTER 6. STATE SPACE DECOMPOSITION

where β^l is the inverse annealing temperature for the layer chosen dynamically for each layer [26] or by a predetermined schedule [14]. In each layer, samples are selected according to their normalized weights. The selected samples are then perturbed by an adaptive diffusion kernel, and re-weighted to result in a normalized set of samples. At the end of the last layer, the expected state of the sample set is considered to be the estimate of the state.

The principal change we made to the procedure is in Eq. (6.11). The sample weights in the modified procedure are obtained by evaluating the annealed marginal likelihoods rather than the overall likelihood. Formally,

$$\log q_i^j = -\beta^l \psi_i(X_i^j), \quad w_i^j = \frac{q_i^j}{\sum_{k=1}^N q_i^k} \quad (6.12)$$

where the subscript i indicates that the sample weights are for the decomposed subspace and ψ_i is the corresponding negative log marginal likelihood obtained by numerical marginalization in Eq. (6.2). Using the decomposed weights, new samples are generated by re-sampling different X_i^j s according to their respective w_i^j s. They are finally combined to produce new samples X^j . This operation is similar to the crossover performed in [26]. However, we perform crossover based on the fit of the individual parts.

6.2.6 An illustrative example

We created a contrived problem to demonstrate the effectiveness of the proposed method. We considered a mixture of four two-dimensional Gaussians shown as a contour plot in Figure 6.5 to be the likelihood. The likelihood is decomposed into a product of 2 factors. We made the search procedure locate the global mode both without decomposition using Eq. (6.11), and with decomposition using Eq. (6.12). In the figure, the proposal samples and the re-sampled ones for the layers 1, 3 and 5 are shown in green and blue respectively. It can be observed that both the methods converge to the maximum likelihood estimate (MLE) in the top-right corner. However, it can be observed that with decomposition the MLE is located quickly and the estimate has a low variance. The example demonstrates that by suitably decomposing the problem, the efficiency of search can be significantly improved, which is further corroborated by the results we present in HMC

6.2. STATE SPACE DECOMPOSITION

Figure 6.5: An illustrative example comparing the current method with the proposed one. Sub-figures in each row show the samples in layers 1, 3, and 5. Proposal samples are shown in green and re-sampled ones in blue (color).

Figure 6.6: Quantitative tracking results for Human Eva I jogging sequences. It can be observed that our method (HD) performs competitively.

using the Human Eva II dataset.

6.3 Experiments and Results

In this section, we first validate the proposed method by comparing it to a baseline configuration using the Human Eva I dataset. Then, we compare the proposed method with the state of the art methods used in HMC [50, 3, 14, 15] using the Human Eva II dataset. For Human Eva I, input from 3 RGB cameras and 2 grayscale cameras was used and for Human Eva II, input from all the 4 cameras was used for tracking. Similar to [3, 15], we registered a set of markers provided by the ground truth for the first frame to the model, in order to measure tracking error. The marker location in subsequent frames were used to measure the error. For Human Eva II, the online evaluation system [15] was used to estimate the tracking error.

We used a kinematic model made of 10 rigid links (L) and 25 DOF for Human Eva I. We used an extended kinematic model, with ankles, for Human Eva II resulting in 12 rigid links and 27 DOF. We used ISA configured with the parameters used in [14] as the baseline procedure for our experiments in the Human Eva I dataset. We used a likelihood based on DDCM (Section 5.2.5). In our experiments, we found that for the torso, the cost ψ_{torso} is poorly constrained. Hence we added the cost corresponding to the parts directly connected to the torso such as the head, upper arms and the legs to ψ_{torso} , in order to perform a stable inference. Figure 6.6 shows the time averaged mean error and one standard deviation of 5 different runs for 12 different sequences in the Human Eva I dataset. In the figure, the baseline algorithm is referred to as **B** and the decomposed search is referred to as **HD**. The number of samples used per layer is displayed next to the name of the configuration for both the algorithms.

It can be noticed that the decomposition significantly improves the performance for sequences such as S1 Jog, S2 TC, S2 Box, S3 Gestures and S3 Box. However, for sequences such as S3 Jog, it can be observed that it performs worse. Furthermore, it can be observed that tracking error is worse when higher number of samples are used for tracking. On analysis, we found that when using decomposed search the model got stuck in incorrect

6.3. EXPERIMENTS AND RESULTS

Figure 6.7: The negative log likelihood (NLL) for the S3 Jogging sequence and the overall tracking results of all the sequences for the six configurations are shown in sub-figures (a) and (b) respectively. Mean error is shown by the asterisk, mean error plus one standard deviation is shown by the bar.

hypotheses. However, the incorrect hypothesis had a higher likelihood. This is observed in Figure 6.7a, which shows the ensemble and time averaged negative log likelihood (NLL) for the S3 Jog sequence. It can be observed that the decomposed search still has a lower cost, i.e., it performs the task of search effectively.

A practical scenario when this happens can be observed from Figure 5.33 (c), which shows a view of the S3 jogging sequence. It can be observed that there is no support for the lower left arm, i.e., there are no edge fragments in the observation corresponding to the lower left arm. However, the synthesized edge fragments would contain them. As a result, if the lower left arm is held in its appropriate pose, the overall cost would be much higher than for example hiding the lower left arm inside the torso. Though having multiple views ameliorates this problem to some extent, due to poor background subtraction even multiple views do not help in some scenarios.

In general we observed that the decomposed search had a very wide effective search volume. Consequently, the decomposed tracker takes a different trajectory in comparison to the baseline version. We believe this results in marginally higher tracking error in sequences such as the S1 Gestures, S1 Box, and S2 Walk. However, these artifacts are caused by aspects such as poor model and observation, rather than the search procedure itself. This claim is further strengthened by the Human Eva II results that we present later which uses an accurate model and a relatively less noisy observation.

The overall mean error and deviation for all the 12 sequences is shown

CHAPTER 6. STATE SPACE DECOMPOSITION

Figure 6.8: Qualitative comparison of the tracked output for the subject S4, frame 580, camera 1. It can be observed that the leg pose estimated by our method is better than those reported in [14].

Figure 6.7b. It can be noticed from the average performance, that both the mean error and the deviation are reduced in comparison to the baseline method. In addition, it can be noticed that the performance of the decomposed search is not significantly affected by the reduction in number of samples as opposed to the baseline method.

We compare our results with [14] for Human Eva II, since studies such as [50, 3] use the implementation of [14]. Furthermore, the study in [15] uses an approximate model and hence it results in much higher tracking error than [14]. Rather than implementing the method in [14] ourselves, we compare our approach to the L1 results reported in [14], which is equivalent to our search procedure. We used the surface mesh provided with the Human Eva II dataset for tracking. Similar to [14], we reduced the mesh to have 4000 triangles to have an acceptable computational load.

A high speed approximate method using KD trees was used to synthesize directional edge fragments from the surface mesh. Figure 6.3b shows the synthesized edge fragments, which can be observed to contain a few incorrect occluded edge fragments. We obtained the probability in Eq. (6.10) from the skinning parameters [82] of the vertex. The distance $d_{\mathcal{P}}$ was robustified with the Geman-McClure function $\rho(x) = \frac{x^2}{\alpha^2 + x^2}$ in order make the objective robust to outliers.

The decomposed search procedure was configured to use 8 layers and 70

6.3. EXPERIMENTS AND RESULTS

samples per layer. The annealing schedule parameter α for the decomposed search was set to 0.2 and the adaptive diffusion parameter γ was set to 0.4. The parameter T described in Section 6.2.4 was set to 15. We used a constant position model [26] for prediction and a simple silhouette extraction method. The observation set O^c included directional edge fragments from the silhouette and gray image in the foreground region. Table 6.1 summarizes the tracking results for the two sequences compared, where our method is referred to as **HD SA** (since it uses silhouette and appearance). It can be observed that our procedure performs better than [14] on most slots.

If the appearance of the subject is highly textured, the noise can be significantly higher than methods such as directional chamfer matching can handle. In such a scenario the silhouette alone is the most reliable image feature. Hence we ran the above procedure with the observation set O^c containing directional edge fragments from the silhouette alone. The results for this test are summarized in Table 6.1 as **HD S** (since it uses only silhouette). It can be observed that even without using appearance related features, our procedure results in comparable tracking performance as [14]. We provide the baseline results using ISA and DDCM with the parameters used in [14], as **B SA** in the table. It can be observed that the proposed method **HD SA** is significantly better than the baseline.

Figures 6.8 shows the tracked result superimposed on the observation from the S4 sequence. It can be observed that the leg pose estimated by our method is slightly better than that in [14]. Figure 6.9 shows the tracked result for the S2 and S4 sequences from the Human Eva II dataset. Video of the tracked and smoothed results obtained using our method is provided with the supplementary material.

Table 6.2 presents the tracking results of our method next to other prominent methods in the current literature. These results, which are extracted from [14], can be directly compared to our results since both are evaluated by the Human Eva II online system. The mean absolute tracking error and deviation for the frames 2 to 821 of the sequence S4 is shown in the table. It can be observed that our method performs better than other prominent methods in the current literature.

Partitioned sampling [118] is a fundamental idea that is used in our

CHAPTER 6. STATE SPACE DECOMPOSITION

Frames		S2			Samples used	Time (sec) per frame
		1-350	1-700	1-1202		
Absolute	[14]	41.5 ± 8.0	45.0 ± 12.9	43.8 ± 10.7	3750	76
$\mu \pm \sigma$	HD SA	37.0 ± 6.9	41.7 ± 9.1	42.4 ± 9.6	560	6
(mm)	HD S	39.4 ± 7.6	44.6 ± 12.6	46.6 ± 12.9	560	6
	B SA	39.2 ± 6.8	44.7 ± 9.8	44.2 ± 9.2	3750	40
Relative	[14]	45.8 ± 9.0	48.4 ± 13.7	46.6 ± 11.4		
$\mu \pm \sigma$	HD SA	41.4 ± 9.0	43.4 ± 8.9	45.2 ± 10.1		
(mm)	HD S	44.8 ± 10.3	47.9 ± 13.1	50.2 ± 14.3		
	B SA	45.0 ± 9.9	48.6 ± 10.7	48.4 ± 10.2		

Frames		S4			Samples used	Time (sec) per frame
		2-350	2-700	2-1258		
Absolute	[14]	34.6 ± 4.6	38.5 ± 6.9	38.1 ± 5.8	3750	76
$\mu \pm \sigma$	HD SA	31.2 ± 5.4	34.8 ± 6.4	36.3 ± 5.7	560	6
(mm)	HD S	31.6 ± 5.1	35.3 ± 6.1	37.1 ± 6.2	560	6
	B SA	32.5 ± 5.9	36.7 ± 7.5	40.7 ± 9.9	3750	40
Relative	[14]	43.9 ± 8.2	47.0 ± 10.6	45.3 ± 9.1		
$\mu \pm \sigma$	HD SA	32.0 ± 5.9	36.2 ± 7.6	38.2 ± 7.4		
(mm)	HD S	32.4 ± 5.7	37.0 ± 7.3	39.0 ± 7.9		
	B SA	32.7 ± 6.5	38.3 ± 9	43.4 ± 13.5		

Table 6.1: Our tracking results for the Human Eva II dataset are presented next to those reported in [14] for the subjects S2 and S4. The absolute and the relative error were obtained using the online evaluation system [15]. Best result in **bold**.

	PF	ICP	PFICP	APF	ISA	HD
	[87]		[140]	[26]	[14]	
μ (error in mm)	104.61	63.86	69.70	44.15	38.58	35.32
σ	40.77	27.07	24.75	15.39	6.54	6.22

Table 6.2: Mean and standard deviation of the absolute tracking error for our method is reported next to other prominent techniques. Results extracted from [14] for S4 (frames 2-821). Best result in **bold**.

Sequence	[118]	[55]	HD
S2	64.8 ± 14.3	78 ± 10	36.6 ± 6.5
S4	71.5 ± 15.3	80 ± 13	29.6 ± 4.8

Table 6.3: Mean and standard deviation of the absolute tracking error for our method is compared to other related methods. Comparison is done on the first 150 frames of S2 and S4 sequence from the Human Eva II dataset. Best result in **bold**.

6.3. EXPERIMENTS AND RESULTS

Figure 6.9: Tracking results for S2 and S4 sequences from Human Eva II. Odd rows show the model output superimposed on the input from the camera, even rows show the 3d model.

method. Table 6.3 compares our method to the basic partitioned sampling scheme. The main difference between partitioned sampling and our inference scheme is that, partitioned sampling uses a single layer with 560 particles opposed to our method which uses 8 layers with 70 particles each. It can be observed that our method performs significantly better than the basic partitioned sampling scheme. This is expected since the annealing technique used in our method is expected to efficiently move the particles towards the modes of the likelihood.

In addition to partitioned sampling, Table 6.3 presents the tracking results for a recent 3D model-free method [55]. Since the results reported in [55] were evaluated by the Human Eva II online system, we extracted the results from [55] rather than implementing their method. It can be observed that our method performs significantly better than the model-free method.

The decomposition procedure we introduce marginally adds to the per

 CHAPTER 6. STATE SPACE DECOMPOSITION

sample overhead, but we found that this is insignificant in comparison to the rest of the processing. Since computational overhead to HMC is directly related to the number of samples used [15], it can be significantly reduced by using our method. The number of samples used for tracking and the computation time on a standard PC for the Human Eva II dataset is shown in Table 6.1. It can be observed from the table that the proposed method uses less than a sixth of the samples used in [14]. The computation time of our method is significantly lower both due to the decomposed search method which requires significantly lower number of samples and the DDCM based likelihood, which can be realized at high speeds. Furthermore, the implementation in [14, 50, 3] uses a GPU based rendering. We did not use GPU acceleration in any form. We believe that by using a GPU based implementation of our method, HMC with the accuracy achieved in [14, 50, 3] would be possible at few frames per second.

6.4 Summary

In this chapter, we described a probabilistic framework to decompose the high dimensional state space of the human motion capture system. We show that by defining conditional likelihood for each limb rather than an overall likelihood for the human model, a number of conditional independence assumptions can be made that enable the decomposition of the state space. We extend the state-of-the-art search method for HMC to make use of the decomposed subspaces. We demonstrate using the Human Eva I and II datasets that the decomposition framework performs competitively when compared to the state of the art system despite using a small fraction of the computational resources. In the next chapter, we discuss the future work that needs to be carried out for a realizing a marker-less motion capture system and provide the concluding remark for the thesis.

Chapter 7

Summary and Future work

7.1 Conclusion

In this thesis, we describe various techniques that improve the state-of-the-art marker-less human motion capture systems. In this chapter, we summarize the novel contributions presented in the thesis. We further discuss the findings of the thesis and indicate how the various ideas described in the thesis can be extended. We also indicate the future research directions that can be pursued in order to achieve the overall objective of the thesis.

In Chapter 4, we describe a stochastic search method that improves on popular methods such as APF [26] and ISA [92]. PA makes two novel steps that improves the performance of stochastic search, namely, reusing samples across annealing layers and using a parametric form to infer the expected state. We demonstrate PA using a contrived example. Furthermore, we present sensitivity analysis and study how the proposed method scales with the dimensionality, multi-modality and the search range of the optimization problem. We compare the proposed method with the current methods such as APF [26] and ISA [92] using the data from the Human Eva I dataset and show that PA outperforms the current methods. Mathematical modeling and analysis of the proposed approach was not pursued in this study. However, such an analysis would provide insights and mathematical rigor to PA.

We also found that good parameters are critical to the performance of

CHAPTER 7. SUMMARY AND FUTURE WORK

the stochastic search methods. Hence, we devised a systematic framework that enables the parameter selection for tracking using stochastic search. We validate our framework using the data from the Human Eva I dataset. The validation used synthetic training data due to the lack of good model and ground truth. The application of the framework using real training data would be an interesting activity that can be pursued in the future.

We further extend the application of stochastic search to the manifold $\mathbb{S}^n$ as opposed to the Euclidean space $\mathbb{R}^n$ in which it is traditionally used. The critical insight is that the stochastic search procedure can be applied to any space with a kernel that can be sampled from and a state estimation procedure. The extension of stochastic search to new manifolds would be interesting. Furthermore, the application of stochastic search to non-convex optimization problems in other general manifolds would also be interesting.

In Chapter 5, we describe a likelihood formulation for human motion capture using a dynamic directional chamfer matching (DDCM) scheme. We adapt the directional chamfer matching scheme used in object recognition for multi-view model-based human motion capture system. We show that the multi-view geometry of the human motion capture system can be exploited to estimate key parameters for the system. We compare the DDCM scheme with the alternatives used in the current systems and show that the proposed likelihood performs significantly better than the alternatives used in state-of-the-art systems.

Our study revealed that the likelihood has the most impact on the tracking accuracy. Our current evaluation is performed using the Human Eva I and II datasets, in which the subjects wear apparel with nearly constant appearance and the background is indoor with substantial background clutter. It would be interesting to apply these techniques with subjects wearing more textured apparel and varying lighting conditions. Furthermore, application of the method to outdoor dataset would be interesting, though we have made some initial foray into the area [20].

In Chapter 6, we describe a framework for human motion capture that enables hierarchical decomposition of the model, thereby significantly reducing the computational overhead required for tracking. We show that rather than defining a joint likelihood, if conditional likelihoods are defined

7.2. FUTURE WORK

for each rigid part that makes the articulated model, the inference procedure can efficiently decompose the high dimensional state space required for representing a human pose. We present a contrived example that demonstrates the effectiveness of the proposed method. We then compare the proposed method with the state-of-the-art technique and show that it is capable of reaching better performance level using less than a sixth of the computational resources.

Though the decomposition results in significant performance improvement, we observe that a good initial estimate for the parameters of the root of the kinematic tree improves the search performance substantially. This can be achieved by performing either a prediction of these parameters or by obtaining the initial estimate through other means such as least squares [14]. This would result in an interesting extension of the proposed method.

We have used APF as the inference technique with decomposition in this thesis, though the stochastic search method PA described in Chapter 4 has shown better results than APF. The main reason for this choice is that, with decomposition there are many marginal likelihoods which are being searched simultaneously. We observe that these marginal likelihoods have different entropy, i.e., some are relatively flat while others are more peaked. Furthermore, the dimensionality and the multi-modality in them are very different. As a result, the optimal parameters for PA for one marginal likelihood is suboptimal for others. Though one could use a different set of parameters for each marginal likelihood, it would result in many parameters to be tuned. We observe that APF is relatively tolerant to such differences. A future research problem would be to identify an annealing schedule that is tolerant to such differences yet enable the reuse of samples across annealing layers.

7.2 Future Work

A number of future research activities were identified during the research in order to make the marker-less motion capture systems robust. They are delineated in this section.

As we indicated in last section, a robust likelihood is the most critical

CHAPTER 7. SUMMARY AND FUTURE WORK

component of the system that enables it to generalize to subjects wearing regular apparel. In order to achieve this with DDCM, reliable extraction of directional edge fragments from the observation is important. We believe that a learning based approach that can discriminate edge fragments from the outline of the subject from those that belong to the background and the clothing of the subject can be used for this purpose. This would result in filtering out the unwanted clutter in the appearance of the subject and extracting reliable edge fragments that are relevant for the likelihood. Furthermore, such a framework would be robust to outdoor scenarios, where the current systems do not function well [95].

We presented a decomposition approach that significantly reduces the computational overhead for tracking. Our experiments for the approach are restricted to single subject. However, the method we present can be scaled efficiently to multiple subjects. Hence the technique can be used in applications such as tracking multiple walking subjects in surveillance videos [4] and tracking the motion of closely interacting subjects in sports videos [50]. The technique can also be applied to tracking other articulated objects such as a human hand. This would significantly reduced the overhead involved in model based hand tracking systems with many degrees of freedom [106].

The decomposed search technique we described is a highly parallel one. It can be accelerated using GPU and parallel processing techniques. Furthermore, fundamental operations used by our method such as KD trees can be realized using a GPU [141]. Hence, we believe that such a realization would result in a high speed method for human motion capture. Furthermore, the features such as directional edge fragments can be extracted at high speed and streamed using a low bandwidth, enabling a cloud based solution for human motion capture. These ideas could not be pursued during the research and it remains a future work.

The decomposed search technique we present in this thesis is a general method that can be applied to other optimization problems as well. It exploits the special structure in the problem to enable efficient inference. Other problem areas that could benefit from such a decomposition could be explored. Furthermore, how it is related to other inference methods such as non parametric belief propagation [75, 136] can be analyzed.

7.2. FUTURE WORK

A drawback of the decomposition method is that it relies on the prior pose to obtain the part labels for the observation. As a result, if there is a tracking failure due to transient observation noise, the likelihood of tracking recovery is reduced. This can be overcome by using bottom-up part detectors or "shouters" used in [41]. Alternatively, part detectors can be obtained using discriminative approaches such as [38, 6]. However, this is not pursued in this thesis and it remains a future work.

Reconstructing human motion using RGBD input such as those obtained from Kinect is a research problem with applications in HCI and gaming. Current solutions rely on discriminative prediction for reinitialization [6] and local optimization based methods such as ICP [55] and optical flow [29] for tracking. In these systems, though local optimization is not robust, any tracking failure can be recovered using reinitialization. One could fuse bottom-up part detectors with model based decomposed search technique as an alternative to obtain a robust system with low computational overhead. The trade-offs involved in such a system needs to be analyzed.

CHAPTER 7. SUMMARY AND FUTURE WORK

Appendices

Appendix A

A.1 $\|A - B\|_a$ is a metric

We show here that $\forall A, B \in [0, \pi)$, the distance function defined as $\|A - B\|_a = \min(|A - B|, \pi - |A - B|)$ is a metric. Since metric properties such as *non-negativity*, *symmetry* and *identity of indiscernibles* ($\|A - B\|_a = 0$, if and only if $A = B$) are trivially true, we focus on the *triangle inequality*. Since the set $[0, \pi)$ modulo π is isomorphic to a circle, without loss of generality, we consider the elements of the set to be elements of the circle S^1 .

We prove triangle inequality $\|A - B\|_a + \|B - C\|_a \geq \|A - C\|_a$ by considering three different situations. Let us consider A and C to be antipodal points on the circle, then equality holds irrespective of B . Let us consider A and C to be any non antipodal points, then the point B is either in the shortest arc or the longest arc subtended by A and C . If B is in the shortest arc, equality holds again. If B is in the longest arc triangle inequality holds. ■

A.2 $d_{\mathcal{P}}(A, B)$ is a metric

We show here that the directional chamfer distance we define in Section 5.2.3 is a metric. Similar to previous proof, since metric properties such as *non-negativity*, *symmetry* and *identity of indiscernibles* are trivially true, we focus on the *triangle inequality*. In order to simplify the proof we use the

APPENDIX A. APPENDIX A

below notation.

$$d_{\mathcal{P}}(A, B) = \sqrt{\frac{d_e^2(A, B)}{\lambda} + d_a^2(A, B)} \quad (\text{A.1})$$

where $d_a(A, B)$ and $d_e(A, B)$ are the orientation and the Euclidean component metrics. In order to show triangle inequality $d_{\mathcal{P}}(A, B) + d_{\mathcal{P}}(B, C) \geq d_{\mathcal{P}}(A, C)$ for $d_{\mathcal{P}}(A, B)$, it suffices to show that the squared triangle inequality below holds

$$d_{\mathcal{P}}^2(A, B) + d_{\mathcal{P}}^2(B, C) + 2d_{\mathcal{P}}(A, B)d_{\mathcal{P}}(B, C) \geq d_{\mathcal{P}}^2(A, C) \quad (\text{A.2})$$

Using the fact that $d_a(A, B)$ and $d_e(A, B)$ are metrics and hence the squared triangle inequality holds for them, we obtain the below equalities

$$d_e^2(A, B) + d_e^2(B, C) + 2d_e(A, B)d_e(B, C) \geq d_e^2(A, C) \quad (\text{A.3})$$

$$d_a^2(A, B) + d_a^2(B, C) + 2d_a(A, B)d_a(B, C) \geq d_a^2(A, C) \quad (\text{A.4})$$

Adding suitably scaled version of the above two inequalities and simplifying, we obtain

$$d_{\mathcal{P}}^2(A, B) + d_{\mathcal{P}}^2(B, C) + \frac{2d_e(A, B)d_e(B, C)}{\lambda} + 2d_a(A, B)d_a(B, C) \geq d_{\mathcal{P}}^2(A, C) \quad (\text{A.5})$$

Comparing the above equation to Eq. (A.2), we observe that it is sufficient to prove the below inequality to prove triangle inequality

$$d_{\mathcal{P}}(A, B)d_{\mathcal{P}}(B, C) \geq \frac{d_e(A, B)d_e(B, C)}{\lambda} + d_a(A, B)d_a(B, C) \quad (\text{A.6})$$

Squaring the above equation and simplifying, we obtain

$$\frac{1}{\lambda} \left(d_a(A, B)d_e(B, C) - d_e(A, B)d_a(B, C) \right)^2 \geq 0 \quad (\text{A.7})$$

The above inequality holds since λ is positive. ■

Appendix B

B.1 Equation 6.2

$$P(lla = l) = \int P(lla = l, ula = u, tor = t) du dt \quad (B.1)$$

The above marginalization can be performed using Monte Carlo approximation, which is explained here. We first explain this for sequential Monte Carlo based methods, then we expand it to stochastic search. Let us assume $G(tor)$, $G(ula)$ and $G(lla)$ to be the prior density, obtained by convolving the posterior from previous time step with a motion prior which typically includes a deterministic drift and stochastic diffusion. G is used as a proposal density for generating samples from $P(lla = l, ula = u, tor = t)$, which is described next. The equation can be expanded as below

$$P(lla) = \int P(lla|ula, tor)P(ula|tor)P(tor) du dt \quad (B.2)$$

One can generate samples from $P(tor)$ by importance sampling using $G(tor)$ as the proposal. Furthermore, since the posterior at the current time step is a product of the prior and the likelihood, it would be sufficient to generate samples from $G(tor)$ and evaluate $P(tor)$. This would result in a weighted unnormalized set of samples that represent the posterior. Furthermore, it can be approximated by a sum of δ functions as below.

$$P(lla) \propto \int P(lla|ula, tor)P(ula|tor) \sum_i \delta(tor_i) \exp^{-\phi_{tor}(tor_i)} du dt \quad (B.3)$$

APPENDIX B. APPENDIX B

We use $\propto$ since the posterior for tor is unnormalized. Applying linearity, one can obtain

$$P(lla) \propto \sum_i \int P(lla|ula_i, tor_i) P(ula_i|tor_i) \delta(tor_i) \exp^{-\phi_{tor}(tor_i)} du dt \quad (B.4)$$

A sample from $P(ula_i|tor_i)$ can be obtained by importance sampling using $G(ula)$ as the proposal to obtain the below expression

$$P(lla) \propto \sum_i \int P(lla|ula_i, tor_i) \delta(ula_i, tor_i) \exp^{-\phi_{ula}(ula_i) - \phi_{tor}(tor_i)} du dt \quad (B.5)$$

A sample from $P(lla_i|ula_i, tor_i)$ can be obtained by importance sampling using $G(lla)$ as the proposal to obtain the below expression

$$P(lla) \propto \sum_i \int \delta(lla_i, ula_i, tor_i) \exp^{-\phi_{lla}(lla_i) - \phi_{ula}(ula_i) - \phi_{tor}(tor_i)} du dt \quad (B.6)$$

After integration, one can obtain the simplified form below

$$P(lla) \propto \sum_i \delta(lla_i) \exp^{-\phi_{lla}(lla_i) - \phi_{ula}(ula_i) - \phi_{tor}(tor_i)} \quad (B.7)$$

The above equation reveals that the unnormalized marginal likelihood can be expressed as a weighted sum of δ functions centered on the values lla_i with certain weights that involve the parent variables. Hence, the marginal likelihood $P(lla)$ can be expressed as below

$$-\ln P(lla) = \psi_{lla} = \phi_{lla} + \phi_{ula} + \phi_{tor} + C \quad (B.8)$$

where C is the normalization constant, and ψ_{lla} is the negative log marginal. The above expression shows that once we have a conditional likelihood defined for each part, the marginal likelihood for the part can be obtained as a product of the conditional likelihoods of all parent links and the conditional likelihood for the part.

The marginal likelihood described above represent the posterior distribution at a time step t . Stochastic search procedures such as APF, use the posterior as the prior for secondary layers, which can be applied with the factorized search as well.

B.2. EQUATION ??

B.2 Equation 6.4

$$P(x|l, c) = \frac{P(l|x, c)}{\sum_x P(l|x, c)} \quad (\text{B.9})$$

The above equation can be shown by first applying Bayes rule to obtain

$$P(x|l, c) = \frac{P(l|x, c)P(x|c)}{P(l|c)} \quad (\text{B.10})$$

The denominator can be further expanded as below

$$P(x|l, c) = \frac{P(l|x, c)P(x|c)}{\sum_x P(l, x|c)} \quad (\text{B.11})$$

which can be further expanded as

$$P(x|l, c) = \frac{P(l|x, c)P(x|c)}{\sum_x P(l|x, c)P(x|c)} \quad (\text{B.12})$$

Assuming an uniform prior over the edge fragments given the camera, one can obtain the below equation

$$P(x|l, c) = \frac{P(l|x, c)}{\sum_x P(l|x, c)} \quad (\text{B.13})$$

B.3 Equation 6.5

$$P(c|l) = \frac{\sum_x P(l|x, c)}{\sum_{x,c} P(l|x, c)} \quad (\text{B.14})$$

The above equation can be shown by first expanding $P(c|l)$ using Bayes rule to obtain

$$P(c|l) = \frac{P(l|c)P(c)}{P(l)} \quad (\text{B.15})$$

The numerator and denominator can be expanded as below

$$P(c|l) = \frac{\sum_x P(l, x|c)P(c)}{\sum_{x,c} P(l, x, c)} \quad (\text{B.16})$$

The numerator and denominator can be further expanded using the product rule as below

$$P(c|l) = \frac{\sum_x P(l|x, c)P(x|c)P(c)}{\sum_{x,c} P(l|x, c)P(x|c)P(c)} \quad (\text{B.17})$$

APPENDIX B. APPENDIX B

Further assuming an uniform prior over the camera in addition to an uniform prior over the edge fragments given the camera, one can obtain

$$P(c|l) = \frac{\sum_x P(l|x, c)}{\sum_{x,c} P(l|x, c)} \quad (\text{B.18})$$

Bibliography

- [1] T. B. Moeslund and E. Granum, "A survey of computer vision-based human motion capture," *CVIU*, vol. 81, pp. 231–268, 2001.
- [2] D. M. Gavrilu, "The visual analysis of human movement: A survey," *CVIU*, vol. 73, pp. 82–98, 1999.
- [3] J. Gall, C. Stoll, E. de Aguiar, C. Theobalt, B. Rosenhahn, and H.-P. Seidel, "Motion capture using joint skeleton tracking and surface estimation," in *CVPR*, pp. 1746–1753, 2009.
- [4] M. Andriluka, S. Roth, and B. Schiele, "Monocular 3d pose estimation and tracking by detection," in *CVPR*, pp. 623–630, 2010.
- [5] A. Basharat, A. Gritai, and M. Shah, "Learning object motion patterns for anomaly detection and improved object detection," in *CVPR*, pp. 1–8, 2008.
- [6] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *CVPR*, pp. 1297–1304, 2011.
- [7] R. Y. Wang and J. Popović, "Real-time hand-tracking with a color glove," *ACM Transactions on Graphics*, vol. 28, no. 3, 2009.
- [8] R. Urtasun, D. J. Fleet, and P. Fua, "3d people tracking with Gaussian process dynamical models," in *CVPR*, pp. 238–245, 2006.
- [9] <http://freemotionfiles.blogspot.com/2009/05/motion-capture-technology.html>. [Online; accessed 19-September-2011].

BIBLIOGRAPHY

-
- [10] T. B. Moeslund, A. Hilton, and V. Krger, "A survey of advances in vision-based human motion capture and analysis," *CVIU*, vol. 104, pp. 90–126, 2006.
 - [11] R. Poppe, "Vision-based human motion analysis: An overview," *CVIU*, vol. 108, pp. 4–18, 2007.
 - [12] A. O. Balan, L. Sigal, and M. J. Black, "A quantitative evaluation of video-based 3d person tracking," in *International Workshop on Performance Evaluation of Tracking and Surveillance*, pp. 349–356, 2005.
 - [13] <http://en.wikipedia.org/wiki/Kinect>. [Online; accessed 19-September-2011].
 - [14] J. Gall, B. Rosenhahn, T. Brox, and H.-P. Seidel, "Optimization and filtering for human motion capture," *IJCV*, vol. 87, pp. 75–92, 2010.
 - [15] L. Sigal, A. O. Balan, and M. J. Black, "HumanEva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion," *IJCV*, vol. 87, pp. 4–27, 2010.
 - [16] P. Kaliamoorthi and R. Kakarala, "Parametric annealing: A stochastic search method for human pose tracking," *Pattern Recognition*, vol. 46, pp. 1501–1510, 2013.
 - [17] P. Kaliamoorthi and R. Kakarala, "Human pose tracking by parametric annealing," in *IEEE Computer Society conference on Computer Vision and Pattern Recognition, 2nd International Workshop on Human Activity Understanding from 3D Data (HAU3D)*, 2012.
 - [18] R. Kakarala, P. Kaliamoorthi, and V. Premachandran, "Three-dimensional bilateral symmetry plane estimation in the phase domain," in *CVPR*, pp. 1–8, 2013.
 - [19] P. Kaliamoorthi and R. Kakarala, "Directional chamfer matching in 2.5 dimensions," in *IEEE Signal Processing Letters (in press)*, pp. 1–4, 2013.
 - [20] P. Kaliamoorthi and R. Kakarala, "A robust segmentation based approach to human pose tracking," in *PSIVT*, pp. 1–8, 2013.

BIBLIOGRAPHY

-
- [21] P. Kaliamoorthi and R. Kakarala, "Probabilistic state spaces decomposition for human motion capture," in *Pattern Recognition (under review)*, pp. 1–8, 2013.
 - [22] C. Sminchisescu and B. Triggs, "Estimating articulated human motion with covariance scaled sampling," *IJRR*, vol. 22, pp. 371–392, 2003.
 - [23] H. Sidenbladh, M. J. Black, and L. Sigal, "Implicit probabilistic models of human motion for synthesis and tracking," in *ECCV*, pp. 784–800, 2002.
 - [24] T. Drummond and R. Cipolla, "Real-time tracking of highly articulated structures in the presence of noisy measurements," *ICCV*, vol. 2, pp. 315–320, 2001.
 - [25] T.-J. Cham and J. M. Rehg, "A multiple hypothesis approach to figure tracking," in *CVPR*, pp. 2239–2245, 1999.
 - [26] J. Deutscher and I. Reid, "Articulated body motion capture by stochastic search," *IJCV*, vol. 61, pp. 185–205, 2005.
 - [27] H. Sidenbladh, M. J. Black, and D. J. Fleet, "Stochastic tracking of 3d human figures using 2d image motion," in *ECCV*, pp. 702–718, 2000.
 - [28] X. Xu and B. Li, "Learning motion correlation for tracking articulated human body with a Rao-Blackwellised particle filter," in *ICCV*, pp. 1–8, 2007.
 - [29] C. Bregler and J. Malik, "Tracking people with twists and exponential maps," in *CVPR*, pp. 8–15, 1998.
 - [30] I.-C. Chang and S.-Y. Lin, "3d human motion tracking based on a progressive particle filter," *Pattern Recognition*, vol. 43, pp. 3621–3635, 2010.
 - [31] L. Raskin, E. Rivlin, and M. Rudzsky, "Using Gaussian process annealing particle filter for 3d human tracking," *EURASIP Journal on Advances in Signal Processing*, 2007.
 - [32] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas, "Discriminative density propagation for 3d human motion estimation," in *CVPR*, pp. 390–397, 2005.

BIBLIOGRAPHY

-
- [33] A. Agarwal and B. Triggs, “3d human pose from silhouettes by relevance vector regression,” in *CVPR*, pp. 882–888, 2004.
 - [34] A. Agarwal and B. Triggs, “A local basis representation for estimating human pose from cluttered images,” in *ACCV*, pp. 50–59, 2006.
 - [35] R. Rosales and S. Sclaroff, “Learning body pose via specialized maps,” in *NIPS*, pp. 1263–1270, 2001.
 - [36] G. Mori and J. Malik, “Estimating human body configurations using shape context matching,” in *ECCV*, pp. 666–680, 2002.
 - [37] R. Poppe, “Evaluating Example-based Pose Estimation: Experiments on the HumanEva Sets,” in *CVPR Workshop on Evaluation of Articulated Human Motion and Pose Estimation (EHuM2)*, 2007.
 - [38] L. Bo and C. Sminchisescu, “Twin Gaussian processes for structured prediction,” *IJCV*, vol. 87, pp. 28–52, 2010.
 - [39] P. F. Felzenszwalb and D. P. Huttenlocher, “Pictorial structures for object recognition,” *IJCV*, vol. 61, pp. 55–79, 2005.
 - [40] L. Sigal, M. Isard, B. H. Sigelman, and M. J. Black, “Attractive people: Assembling loose-limbed models using non-parametric belief propagation,” in *NIPS*, pp. 1539–1546, 2003.
 - [41] L. Sigal, S. Bhatia, S. Roth, M. J. Black, and M. Isard, “Tracking loose-limbed people,” *CVPR*, vol. 1, pp. 421–428, 2004.
 - [42] B. Wu and R. Nevatia, “Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors,” in *ICCV*, pp. 90–97, 2005.
 - [43] D. Ramanan and D. A. Forsyth, “Finding and tracking people from the bottom up,” in *CVPR*, pp. 467–474, 2003.
 - [44] D. Ramanan, “Learning to parse images of articulated bodies,” *NIPS*, pp. 1129–1136, 2006.
 - [45] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Inc., 1994.

BIBLIOGRAPHY

-
- [46] C. W. Chu, O. C. Jenkins, and M. J. Mataric, "Markerless kinematic model and motion capture from volume sequences," in *CVPR*, pp. 475–482, 2003.
 - [47] D. Anguelov, D. Koller, H.-C. Pang, P. Srinivasan, and S. Thrun, "Recovering articulated object models from 3d range data," in *UAI*, pp. 18–26, 2004.
 - [48] N. Hasler, T. Thormählen, B. Rosenhahn, and H.-P. Seidel, "Learning skeletons for shape and pose," in *ACM SIGGRAPH*, pp. 23–30, 2010.
 - [49] A. O. Balan, L. Sigal, M. J. Black, J. E. Davis, and H. W. Haussecker, "Detailed human shape and pose from images," in *CVPR*, pp. 1–8, 2007.
 - [50] Y. Liu, C. Stoll, J. Gall, H.-P. Seidel, and C. Theobalt, "Markerless motion capture of interacting characters using multi-view image segmentation," in *CVPR*, pp. 1249–1256, 2011.
 - [51] L. Sigal, A. O. Balan, and M. J. Black, "Combined discriminative and generative articulated pose and non-rigid shape estimation," in *NIPS*, pp. 1337–1344, 2007.
 - [52] M. Andriluka, S. Roth, and B. Schiele, "People-tracking-by-detection and people-detection-by-tracking," in *CVPR*, pp. 1–8, 2008.
 - [53] J. Deutscher, A. Blake, and I. Reid, "Articulated body motion capture by annealed particle filtering," in *CVPR*, vol. 2, pp. 126–133, 2000.
 - [54] J. Deutscher, A. Davison, and I. Reid, "Automatic partitioning of high dimensional search spaces associated with articulated body motion capture," in *CVPR*, vol. 2, pp. 669–676, 2001.
 - [55] S. Corazza, L. Mündermann, E. Gambaretto, G. Ferrigno, and T. P. Andriacchi, "Markerless motion capture through visual hull, articulated icp and subject specific model generation," *International journal of computer vision*, vol. 87, pp. 156–169, 2010.
 - [56] L. Bo and C. Sminchisescu, "Twin Gaussian processes for structured prediction," *IJCV*, vol. 87, pp. 28–52, 2010.

BIBLIOGRAPHY

-
- [57] G. Shakhnarovich, P. Viola, and T. Darrell, "Fast pose estimation with parameter-sensitive hashing," in *ICCV*, pp. 750–759, 2003.
 - [58] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *TPAMI*, vol. 24, pp. 509–522, 2002.
 - [59] D. G. Lowe, "Object recognition from local scale-invariant features," in *ICCV*, pp. 1150–1157, 1999.
 - [60] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, pp. 886–893, 2005.
 - [61] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, pp. 21–27, 1967.
 - [62] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *Journal of Machine Learning Research*, vol. 1, pp. 211–244, 2001.
 - [63] S. E. Yuksel, J. N. Wilson, and P. D. Gader, "Twenty years of mixture of experts," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, pp. 1177–1193, 2012.
 - [64] B. Schölkopf and C. J. Burges, *Advances in kernel methods: support vector learning*. The MIT press, 1999.
 - [65] A. Criminisi, J. Shotton, and E. Konukoglu, "Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning," *Foundations and Trends in Computer Graphics and Vision*, vol. 7, pp. 81–227, 2012.
 - [66] M. A. Fischler and R. A. Elschlager, "The representation and matching of pictorial structures," *IEEE Transactions on Computers*, vol. 22, pp. 67–92, 1973.
 - [67] M. Burenius, J. Sullivan, and S. Carlsson, "3d pictorial structures for multiple view articulated pose estimation," in *CVPR*, pp. 1–8, 2013.
 - [68] Y. Freund and R. E. Schapire, "A short introduction to boosting," in *International Joint Conference on Artificial Intelligence*, pp. 1401–1406, 1999.

BIBLIOGRAPHY

-
- [69] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *CVPR*, vol. 1, pp. 511–518, 2001.
 - [70] E. B. Sudderth, A. T. Ihler, M. Isard, W. T. Freeman, and A. S. Willsky, "Nonparametric belief propagation," *Communications of the ACM*, vol. 53, pp. 95–103, 2010.
 - [71] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009.
 - [72] D. Ramanan and C. Sminchisescu, "Training deformable models for localization," in *CVPR*, pp. 206–213, 2006.
 - [73] M. Isard, "PAMPAS: real-valued graphical models for computer vision," in *CVPR*, pp. 613–620, 2003.
 - [74] D. W. Marquardt, "An algorithm for least-squares estimation of non-linear parameters," *Journal of the Society for Industrial & Applied Mathematics*, vol. 11, pp. 431–441, 1963.
 - [75] L. Sigal, M. Isard, H. W. Haussecker, and M. J. Black, "Loose-limbed people: Estimating 3d human pose and motion using non-parametric belief propagation," *IJCV*, vol. 98, pp. 15–48, 2012.
 - [76] D. Ramanan, D. A. Forsyth, and A. Zisserman, "Tracking people by learning their appearance," *TPAMI*, vol. 29, pp. 65–81, 2007.
 - [77] B. Wu and R. Nevatia, "Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors," *IJCV*, vol. 75, pp. 247–266, 2007.
 - [78] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: Real-time tracking of the human body," *TPAMI*, vol. 19, pp. 780–785, 1997.
 - [79] I. Haritaoglu, D. Harwood, and L. S. Davis, "W4: Who? when? where? what? a real time system for detecting and tracking people," in *IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 222–227, 1998.

BIBLIOGRAPHY

-
- [80] C. Bregler, J. Malik, and K. Pullen, "Twist based acquisition and tracking of animal and human kinematics," *IJCV*, vol. 56, pp. 179–194, 2004.
 - [81] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.
 - [82] L. Ballan and G. M. Cortelazzo, "Marker-less motion capture of skinned models in a four camera set-up using optical flow and silhouettes," in *3DPVT*, June 2008.
 - [83] L. Ballan, A. Taneja, J. Gall, L. V. Gool, and M. Pollefeys, "Motion capture of hands in action using discriminative salient points," in *ECCV*, pp. 640–653, 2012.
 - [84] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis, "SCAPE: shape completion and animation of people," *ACM TOG*, vol. 24, pp. 408–416, 2005.
 - [85] N. Hasler, C. Stoll, M. Sunkel, B. Rosenhahn, and H.-P. Seidel, "A statistical model of human pose and body shape," in *Computer Graphics Forum (Proc. Eurographics 2008)*, vol. 2, 2009.
 - [86] Z. Zhang and H. S. Seah, "Real-time tracking of unconstrained full-body motion using niching swarm filtering combined with local optimization," in *Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 23–28, 2011.
 - [87] M. Isard and A. Blake, "Condensation - conditional density propagation for visual tracking," *IJCV*, vol. 29, pp. 5–28, 1998.
 - [88] K. Choo and D. J. Fleet, "People tracking using hybrid monte carlo filtering," in *ICCV*, 2001.
 - [89] E. Poon and D. J. Fleet, "Hybrid monte carlo filtering: Edge-based people tracking," in *Proceedings of the Workshop on Motion and Video Computing*, pp. 151–158, 2002.
 - [90] R. M. Neal, "MCMC using Hamiltonian dynamics." *Handbook of Markov Chain Monte Carlo*, Chapman and Hall/CRC Press, 2010.

BIBLIOGRAPHY

-
- [91] A. Doucet, N. de Freitas, K. Murphy, and S. Russell, "Rao-blackwellised particle filtering for dynamic bayesian networks," in *UAI*, pp. 176–183, 2000.
 - [92] J. Gall, J. Potthoff, C. Schnrr, B. Rosenhahn, and H.-P. Seidel, "Interacting and annealing particle filters: Mathematics and a recipe for applications," *Journal of Mathematical Imaging and Vision*, pp. 1–18, 2007.
 - [93] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.
 - [94] N. Hasler, B. Rosenhahn, T. Thormhlen, M. W, J. Gall, H. peter Seidel, and M. Informatik, "Markerless motion capture with unsynchronized moving cameras," in *CVPR*, 2009.
 - [95] G. Pons-Moll, A. Baak, J. Gall, L. Leal-Taixé, M. Müller, H.-P. Seidel, and B. Rosenhahn, "Outdoor human motion capture using inverse kinematics and von Mises-Fisher sampling," in *ICCV*, pp. 1243–1250, 2011.
 - [96] A. O. Bălan, M. J. Black, L. Sigal, and H. W. Houssecker, "Shining a light on human pose: On shadows, shading and the estimation of pose and shape," in *ICCV*, pp. 1–8, 2007.
 - [97] P. Guan, A. Weiss, A. O. Balan, and M. J. Black, "Estimating human shape and pose from a single image," in *ICCV*, pp. 1381–1388, 2009.
 - [98] M. A. Brubaker, D. J. Fleet, and A. Hertzmann, "Physics-based person tracking using the anthropomorphic walker," *IJCV*, vol. 87, pp. 140–155, 2010.
 - [99] I. Rius, J. González, J. Varona, and F. X. Roca, "Action-specific motion prior for efficient bayesian 3d human body tracking," *Pattern Recognition*, vol. 42, pp. 2907–2921, 2009.
 - [100] R. Urtasun, D. J. Fleet, and P. Fua, "3d people tracking with Gaussian process dynamical models," in *CVPR*, pp. 238–245, 2006.
 - [101] J. M. Wang, D. J. Fleet, and A. Hertzmann, "Gaussian process dynamical models," in *NIPS*, pp. 1441–1448, 2005.

BIBLIOGRAPHY

-
- [102] N. D. Lawrence, "Gaussian process latent variable models for visualisation of high dimensional data," in *NIPS*, pp. 329–336, 2003.
 - [103] V. John, E. Trucco, and S. Ivezic, "Markerless human articulated tracking using hierarchical particle swarm optimisation," *Image and Vision Computing*, vol. 28, pp. 1530–1547, 2010.
 - [104] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *IEEE International Conference on Neural Networks*, pp. 1942–1948, 1995.
 - [105] I. Oikonomidis, N. Kyriazis, and A. A. Argyros, "Full dof tracking of a hand interacting with an object by modeling occlusions and physical constraints," in *ICCV*, pp. 2088–2095, 2011.
 - [106] I. Oikonomidis, N. Kyriazis, and A. A. Argyros, "Tracking the articulated motion of two strongly interacting hands," in *CVPR*, pp. 1862–1869, 2012.
 - [107] X. Zhao and Y. Liu, "Generative tracking of 3d human motion by hierarchical annealed genetic algorithm," *Pattern Recognition*, vol. 41, pp. 2470–2483, 2008.
 - [108] P. Kohli, J. Rihan, M. Bray, and P. H. Torr, "Simultaneous segmentation and pose estimation of humans using dynamic graph cuts," *IJCV*, vol. 79, pp. 285–298, 2008.
 - [109] Y. Boykov and G. Funka-Lea, "Graph cuts and efficient n-d image segmentation," *IJCV*, vol. 70, pp. 109–131, 2006.
 - [110] C. Stoll, N. Hasler, J. Gall, H.-P. Seidel, and C. Theobalt, "Fast articulated motion tracking using a sums of Gaussians body model," in *ICCV*, pp. 951–958, 2011.
 - [111] A. O. Balan and M. J. Black, "The naked truth: Estimating body shape under clothing," in *ECCV*, pp. 15–29, 2008.
 - [112] R. G. Brown and P. Y. C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering*, vol. 3rd ed. Wiley, 1997.
 - [113] G. Welch and G. Bishop, "An introduction to the kalman filter," 1995.

BIBLIOGRAPHY

-
- [114] Z. H. E. Chen, "Bayesian filtering: From Kalman filters to particle filters , and beyond," *Statistics*, pp. 1–69, 2003.
 - [115] www.cs.nyu.edu/~roweis/notes/gaussid.pdf. [Online; accessed 19-September-2011].
 - [116] D. J. C. MacKay, "Introduction to Monte Carlo methods," in *Proceedings of the NATO Advanced Study Institute on Learning in Graphical Models*, pp. 175–204, 1998.
 - [117] P. Djuric, J. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. Bugallo, and J. Miguez, "Particle filtering," *IEEE Signal Processing Magazine*, pp. 19–38, 2003.
 - [118] J. Maccormick and M. Isard, "Partitioned sampling, articulated objects, and interface-quality hand tracking," in *ECCV*, pp. 3–19, 2000.
 - [119] R. M. Neal, "Annealed importance sampling," *Statistics and Computing*, vol. 11, pp. 125–139, 2001.
 - [120] E. Cantú-Paz, "A survey of parallel genetic algorithms," *Calculateurs paralleles, reseaux et systems repartis*, vol. 10, pp. 141–171, 1998.
 - [121] J. Deutscher, M. Isard, and J. MacCormick, "Automatic camera calibration from a single manhattan image," in *ECCV*, pp. 175–205, 2002.
 - [122] B. Daubney and X. Xie, "Tracking 3d human pose with large root node uncertainty," in *CVPR*, pp. 1321–1328, 2011.
 - [123] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., 2006.
 - [124] M. A. Carreira-Perpiñán, "Mode-finding for mixtures of Gaussian distributions," *TPAMI*, vol. 22, pp. 1318–1323, 2000.
 - [125] D. M. Blei and M. I. Jordan, "Variational inference for dirichlet process mixtures," *Bayesian Analysis*, vol. 1, pp. 121–144, 2005.
 - [126] N. J. B. McFarlane and C. P. Schofield, "Segmentation and tracking of piglets in images," *Machine Vision and Applications*, vol. 8, pp. 187–193, 1995.
 - [127] M. A. Abramson, "Pattern Search Filter Algorithms for Mixed Variable General Constrained Optimization Problems," *Ph.D. Thesis, De-*

BIBLIOGRAPHY

partment of Computational and Applied Mathematics, Rice University, August 2002.

- [128] M. A. Osborne, R. Garnett, and S. J. Roberts, “Gaussian processes for global optimization,” in *Learning and Intelligent Optimization (LION)*, pp. 1–15, 2009.
- [129] http://en.wikipedia.org/wiki/Von_Mises%E2%80%93Fisher_distribution. [Online; accessed 2-July-2013].
- [130] R. Kakarala and D. Mao, “A theory of phase-sensitive rotation invariance with spherical harmonic and moment-based representations,” in *CVPR*, pp. 105–112, 2010.
- [131] R. Kakarala, P. Kaliamoorthi, and W. Li, “Viewpoint invariants from three dimensional data: the role of reflection in human activity understanding,” in *Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 57–62, 2011.
- [132] A. T. Wood, “Simulation of the von mises fisher distribution,” *Communications in statistics-simulation and computation*, vol. 23, pp. 157–164, 1994.
- [133] S. Sra, “A short note on parameter approximation for von mises-fisher distributions: and a fast implementation of $I_s(x)$,” *Computational Statistics*, vol. 27, pp. 177–190, 2012.
- [134] J. Shotton, A. Blake, and R. Cipolla, “Multiscale categorical object recognition using contour fragments,” *TPAMI*, vol. 30, pp. 1270–1281, 2008.
- [135] M.-Y. Liu, O. Tuzel, A. Veeraraghavan, and R. Chellappa, “Fast directional chamfer matching,” in *CVPR*, pp. 1696–1703, 2010.
- [136] E. B. Sudderth, M. I. Mandel, W. T. Freeman, and A. S. Willsky, “Visual hand tracking using nonparametric belief propagation,” in *IEEE Workshop on Generative Model Based Vision*, p. 189, 2004.
- [137] D. M. Gavrilu and L. S. Davis, “3-d model-based tracking of humans in action: a multi-view approach,” in *CVPR*, pp. 73–80, 1996.

BIBLIOGRAPHY

- [138] https://en.wikipedia.org/wiki/K-d_tree. [Online; accessed 5-August-2013].
- [139] E. B. Sudderth, A. T. Ihler, W. T. Freeman, and A. S. Willsky, “Non-parametric belief propagation,” in *CVPR*, pp. 605–612, 2003.
- [140] M. Bray, E. Koller-Meier, and L. Van Gool, “Smart particle filtering for high-dimensional tracking,” *CVIU*, vol. 106, pp. 116–129, 2007.
- [141] K. Zhou, Q. Hou, R. Wang, and B. Guo, “Real-time kd-tree construction on graphics hardware,” in *ACM SIGGRAPH Asia*, vol. 27, pp. 1–11, 2008.

BIBLIOGRAPHY

List of journal publications

- [1] P. Kaliamoorthi and R. Kakarala, "Parametric annealing: A stochastic search method for human pose tracking," *Pattern Recognition*, vol. 46, no. 5, pp. 1501–1510, May 2013.
- [2] P. Kaliamoorthi and R. Kakarala, "Directional chamfer matching in 2.5 dimensions," *IEEE Signal Processing Letters*, vol. 20, no. 12, pp. 1151–1154, Dec 2013. (This paper was chosen from among 83 papers for the cover photo of printed volume 20)
- [3] P. Kaliamoorthi and R. Kakarala, "Probabilistic state spaces decomposition for human motion capture," *Pattern Recognition (under review)*, pp. 1–8, 2013.

LIST OF JOURNAL PUBLICATIONS

List of conference publications

- [1] P. Kaliamoorthi and R. Kakarala, "A robust segmentation based approach to human pose tracking," in *PSIVT*, pp. 1–8, 2013.
- [2] R. Kakarala, P. Kaliamoorthi, and V. Premachandran, "Three-dimensional bilateral symmetry plane estimation in the phase domain," in *CVPR*, pp. 1–8, 2013.
- [3] P. Kaliamoorthi and R. Kakarala, "Human pose tracking by parametric annealing," in *IEEE Computer Society conference on Computer Vision and Pattern Recognition, 2nd International Workshop on Human Activity Understanding from 3D Data (HAU3D)*, 2012.
- [4] R. Kakarala, P. Kaliamoorthi, and W. Li, "Viewpoint invariants from three dimensional data: the role of reflection in human activity understanding," in *Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 57–62, 2011.

LIST OF CONFERENCE PUBLICATIONS