HUMAN MOTION ANALYSIS: FROM GAIT MODELING TO SHAPE REPRESENTATION AND POSE ESTIMATION

By

MENG DING

Bachelor of Science in Electrical Engineering
Beijing Institute of Technology
Beijing, China
2007

Master of Science in Optoelectronic Engineering
Beijing Institute of Technology
Beijing, China
2009

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
DOCTOR OF PHILOSOPHY
July, 2015

HUMAN MOTION ANALYSIS: FROM GAIT MODELING TO SHAPE REPRESENTATION AND POSE ESTIMATION

Dissertation Approved:

Dr. Guoliang Fan

Dissertation Advisor

Dr. Martin Hagan

Dr. Weihua Sheng

Dr. Yi Fang

Dr. R. Russell Rhinehart

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to all those who have helped me to make this dissertation possible.

First of all, I would like to deeply acknowledge my advisor Dr. Guoliang Fan for his invaluable support, guidance and encouragement during my PhD study. His insightful and inspirational suggestions have helped me greatly on my research work. His enthusiasm and dedication for research always encourage me to move forward and overcome many difficulties. I will not forget his hard work for the revision of each of our conference and journal papers, and also I will not forget every exciting and cheering moment with Dr. Fan when our papers are accepted. I feel extremely lucky to work with him and learn from him.

I would also like to thank all of my committee members: Dr. Martin Hagan, Dr. Weihua Sheng, Dr. R. Russell Rhinehart and Dr. Yi Fang for their time, efforts and helpful guidance. Their comments on my research are very insightful, and it is my great honor to discuss with them.

Many thanks to all of my colleagues in the VCIPL who have contributed to my personal and professional time at OSU. I want to thank Song Ge and Liangjiang Yu for working together and spending five years in the lab. I am thankful to Xin Zhang, Jiulu Gong, Ayesha Siddiqua, Brian Yin, Mohammad Nazmul Khan, Jie Yuan and Yi Ding for valuable discussions and sharing colorful life in the lab. I would also thank all of my friends in Stillwater whose company, understanding and wishes made my life here memorable.

Last but not the least, I would like to express my deepest gratitude to my parents,

Xinlin Ding and Chunling Fang, for their unconditional love, continuous support and consecutive encouragement in the past thirty years. Without their generosity, understanding and sacrifice, I cannot complete my Ph.D. study. I am also grateful to my girlfriend, Shenglan He, whose dedication and kindness make my life full of happiness and love. When I look back on the past five years, I experienced countless challenges; however, all of them stimulated my growth and made me stronger.

This work is supported by the Oklahoma Center for the Advancement of Science and Technology (OCAST) under Grants HR09-30, HR12-30 and the National Science Foundation (NSF) under Grant NRI-1427345.

Acknowledgements reflect the views of the author and are not endorsed by committee members or Oklahoma State University.

Name: MENG DING

Date of Degree: July, 2015

Title of Study: HUMAN MOTION ANALYSIS: FROM GAIT MODELING TO SHAPE REPRESENTATION AND POSE ESTIMATION

Major Field: ELECTRICAL ENGINEERING

Abstract: This dissertation presents a series of fundamental approaches to the human motion analysis from three perspectives, i.e., manifold learning-based gait motion modeling, articulated shape representation and efficient pose estimation. Firstly, a new joint gait-pose manifold (JGPM) learning algorithm is proposed to jointly optimize the gait and pose variables simultaneously. To enhance the representability and flexibility for complex motion modeling, we also propose a multi-layer JGPM that is capable of dealing with a variety of walking styles and various strides. We resort to a topologically-constrained Gaussian process latent variable model (GPLVM) to learn the multi-layer JGPM where two new techniques are introduced to facilitate model learning. First is training data diversification that creates a set of simulated motion data with different strides under limited data. Second is the topology-aware local learning that is to speed up model learning by taking advantage of the local topological structure. We demonstrate the effectiveness of our approach by synthesizing the high-quality motions from the multi-layer model. The experimental results show that the multi-layer JGPM outperforms several existing GPLVM-based models in terms of the overall performance of motion modeling.

On the other hand, to achieve efficient human pose estimation from a single depth sensor, we develop a generalized Gaussian kernel correlation (GKC)-based framework which supports not only body shape modeling, but also articulated pose tracking. We first generalize GKC from the univariate Gaussian to the multivariate one and derive a unified GKC function that provides a continuous and differentiable similarity measure between a template and an observation, both of which are represented by a collection of univariate and/or multivariate Gaussian kernels. Then, to facilitate the data matching and accommodate articulated body deformation, we embed a quaternion-based articulated skeleton into a collection of multivariate Gaussians-based template model and develop an articulated GKC (AGKC) which supports subject-specific shape modeling and articulated pose tracking for both the full-body and hand. Our tracking algorithm is simple yet effective and computationally efficient. We evaluate our algorithm on two benchmark depth datasets. The experimental results are promising and competitive when compared with state-of-the-art algorithms.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Motivation

Automatic analysis of human motion is a highly active research topic in the field of computer vision and machine learning due to its wide promising applications, such as surveillance, 3D character animation, biomechanics, robotics, and human-computer interaction (HCI), etc. A few examples are shown in Fig. 1.1. The applications of smart surveillance cover many classical problems for security purposes, like automatically detecting and tracking human motion, access control, people counting, activities analysis and gait recognition. Different with the traditional surveillance system which can only record video and provide "after" evidences, a smart surveillance system with the help of intelligent human motion analysis algorithms, can automatically detect specific events and alarm at real-time. Some examples of the smart surveillance system are illustrated in Fig. 1.1 (a).

In the traditional character animation, we have to record all the motions of a character and embed a virtual human shape or avatar with the recorded motion to generate a animation sequence. However, the task of animating characters can be simplified by using a generative human motion model. In other words, given a motion prior model, plausible poses and motions can be simulated automatically as shown in Fig. 1.1 (b). In biomechanics and medical field, the motion analysis of body parts can help to automatically diagnose orthopedic patients according to some biomechanics data, like body joint angles or the pattern of central of mass, as shown in Fig. 1.1 (c). It can also facilitate in sports to optimize athletic performance or

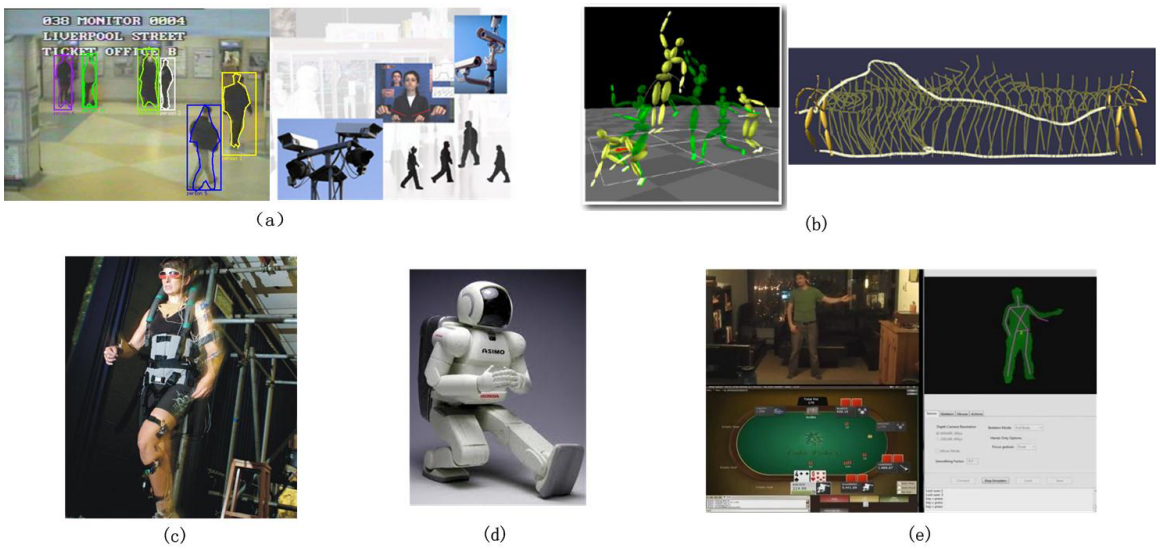


Figure 1.1: A few applications of human motion analysis. (a) Visual Surveillance[1, 2], (b) Character Animation [3, 4],(c) Biomechanics [5],(d) Humanoid Robot [6], (e) Kinect Sensor for HCI [7].

to identify motions that may cause injury or strain. In manufacturing industry, the human motion analysis can be used to control robots or to train the humanoid robots for executing some complicated tasks, as shown in Fig. 1.1 (d). In human-computer interaction (HCI), the estimated motion or pose parameters can be the inputs of computer or video games. Kinect sensor from the Microsoft is a typical example to illustrate that human motion analysis can be employed in the HCI field as shown in Fig. 1.1 (e).

Human motion analysis usually relies on accurate motion capture techniques to collect the human kinematical data. However, there is no one motion capture system can handle all kinds of the human motion with high accuracy in any environment. Especially in video-based posture estimation, due to the high-dimensionality and variability of the motion data as well as the ambiguity from 2D imaging, it is challenging to reconstruct the optimal 3D posture in a high dimensional parameter space. In order to estimate the articulated human motion more accurately and robustly, a mo-

tion model learned from a large-scale training dataset is often used as a statistical prior to constrain the posture search in the solution space. On the other hand, the recent launch of low-cost RGBD sensors (e.g. Kinect) has triggered a large amount of research on human pose estimation. Since depth sensors can measure the depth information and provide a 2.5D depth image at real-time, they have effectively simplified the task of foreground / background substraction and significantly reduced pose ambiguities in monocular human pose estimation. Fig. 1.2 shows (a) two versions of Kinect sensors, (b) the 2.5D depth map captured by Kinect V2 and (c) its corresponding 3D point cloud reconstructed from the depth map.

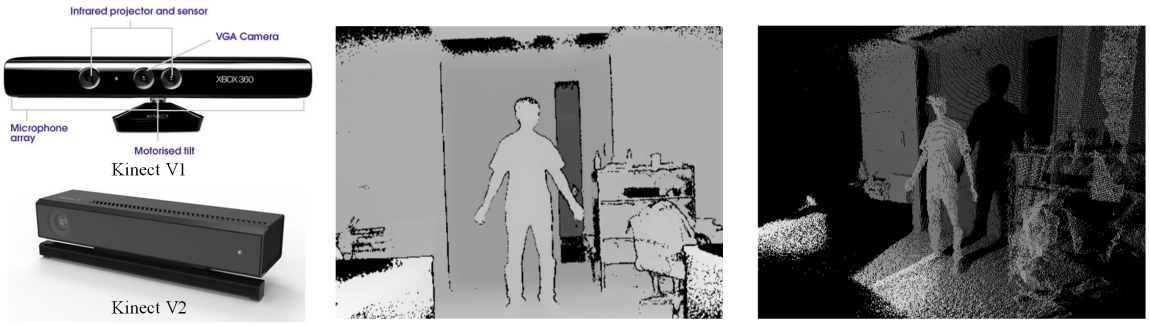


Figure 1.2: (a) Kinect sensors [8, 9], (b) The 2.5D depth map captured by Kinect V2,(c) The corresponding 3D point cloud.

In this research, we focus on the human motion analysis from three aspects, i.e., gait modeling, articulated shape representation and efficient articulated pose estimation. Gait is defined as "a manner of walking" in the Webster's New Collegiate Dictionary. Through finding human walking pattern, gait analysis assess all kinds of underling bio-mechanical characteristics, by which the walking ability of humans can be evaluated. This research can be extended and lead to a few unique applications, such as medical diagnostics, rehabilitation medicine, biometric identification, sport training and fall-risk assessment etc., shown in Fig. 1.3. The key technique of video-based gait analysis is gait modeling from existing motion data. Particularly,

(a) (b) (c)

Figure 1.3: Applications of human gait analysis. (a) Rehabilitation [10], (b) Gait Identification [11], (c) Fall-risk Assessment [12].

we learn a statistical representative low dimensional motion prior from a series of high dimensional gait data to constrain the motion reconstruction in the solution space. Then, this statistical prior model can be employed in video-based gait estimation, computation of center of mass (COM) of body, motion filtering and animation synthesis.

The second aspect in this research is articulated shape representation. A good shape model not only captures shape variability accurately, but also facilitates the pose estimation effectively and efficiently. One of the most widely used shape models is the mesh surface which can depict the object precisely. Good mesh models, which are usually collected by one or multiple high-cost 3D scanners, are difficult to be accessed and even harder to be specified for each subject. Also, when using detailed mesh models for human pose estimation, there always involves a relatively high computational load and the real-time performance can hardly be achieved only using CPU, even if the pose estimation is based on the depth sensor. Moreover, the complicated deformation and blending between mesh surface and the skeleton has to be considered when to have an articulated shape model. These challenges motivate us to develop a simple yet effective parametric shape representation which can support efficient articulated pose estimation. Also, this general shape representation can be easily deformed to match a subject-specific body shape.

The third aspect is efficient articulated pose estimation from monocular depth sensor, which has become a highly active research topic in the computer vision field due to its simplicity for use, low-cost, high efficiency and robustness for the human motion analysis. Many interesting applications based on the motion capture from depth sensors have released, such as natural body controller in video games, home rehabilitation solution for stroke victims and virtual clothes fitting, which are shown in Fig. 1.4.

(a) (b) (c)

Figure 1.4: Applications based on the motion capture from depth sensors. (a) Body controller in video games [13] (b) Home rehabilitation solution for stroke victims [14], (c) virtual clothes fitting [15].

Currently, most of the methods rely on a large scale database for training model or detailed mesh model or both of them. They usually require the acceleration from GPU to achieve the real-time performance, which limits their implementation in mobile device and small embedded system. The current limitations inspire us to develop an efficient and accurate motion capture system from a single depth sensor without using any database nor mesh model. We address the articulated pose estimation through a novel generalized Gaussian kernel correlation function, that is the pose parameters (joint angles between two body segments) are estimated by maximizing a continuous and differentiable Gaussian kernel correlation function with additional constraints. This efficient pose estimation can be extended to hand motion and further support dynamic motion analysis in all kinds of applications, such as biomechanics, medical

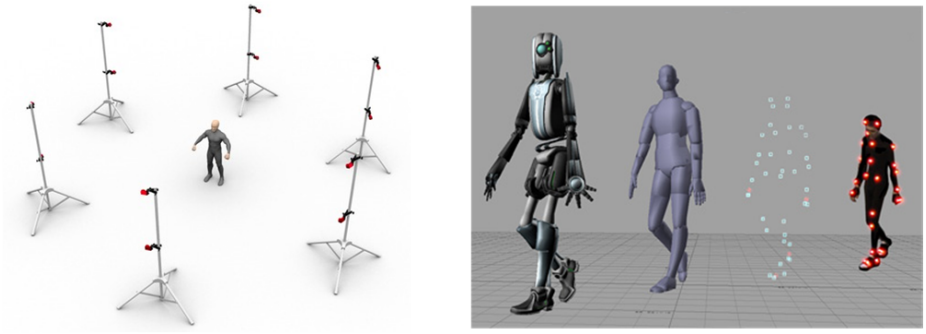
diagnostics and sport training.

1.2 Research Background

Human motion analysis relies on an accurate motion capture (Mocap) system to collect the kinematic motion data represented by Euler angles or 3D positions of the body joints. Nowadays, three kinds of motion capture techniques are most widely used, i.e., marker-based Mocap, markerless Mocap and inertial sensor-based Mocap. While the marker-based Mocap is the *golden standard* in the industry field, the markerless Mocap becomes an active research topic recently and has great potential and promising applications. The selection of a particular Mocap system depends on the requirements and environment of applications.

1.2.1 Marker-based Mocap

The marker-based motion capture system is normally stepped in a laboratory environment. It includes multiple calibrated cameras, a set of markers attached on human body and advanced post-processing software, as shown in Fig. 1.5. When a person performs a series of movements, cameras record and extract the position of each marker in 2D images and the body configuration can be recovered through inverse kinematics (IK) algorithms. The well-known commercial marker-based Mocap system includes Vicon [32] and OptiTrack [33], etc.. The marker-based Mocap system is the most commonly used approach in the industry field due to its high accuracy and robustness. However, it has a few limitations, such as high-cost equipments, specific environment, time consuming preparation and marker attachment, which causes the infeasibility in many applications.



(a) (b)

Figure 1.5: Illustration of the marker-based Mocap. (a) Camera setup in OptiTrack [16] (b) Motion capture and animation demo [17].

1.2.2 Markerless Mocap

Markerless Mocap utilize one or more color cameras to directly calculate the body joint positions from a sequence of images. The markerless motion capture technique allows non-invasive human movement measurement in a natural environment without any marker attachment. Eliminating markers can expand the applicability of human motion capture techniques, considerably reduce the preparation time, and enable simple and accurate motion measurement and assessment in all kinds of applications. Currently, the main markerless Mocap method is the video-based human motion estimation with monocular camera or multiple camera studio. Using one or multiple color cameras for the motion capture has been implemented in a specific laboratory environment. However, it requires complex background substraction procedure and high computation load, making it difficult to run in real-time. Also, the existing ambiguity problem from the 2D images makes the system not accurate and robust for practical usage. Nowadays, the depth sensor-based Mocap is more and more popular due to its low cost and high performance. Since depth sensors can measure the depth information and provide a 2.5D depth image at real-time, they have effectively simplified the task of foreground / background substraction and significantly reduced

pose ambiguities in monocular human pose estimation.

1.2.3 Inertial Sensor-based Mocap

Inertial sensors can measure the 3D rotation angles at body joints when they are mounted near a set of body joints. Then the collected joint angles can reconstruct the full body skeleton through a set of conversion of coordination systems along a chain structure. Compared with the marker and markerless Mocap system, inertial sensor-based Mocap is more accurate on the measurement of joint angle, meanwhile it is not constrained by the application environment. However, inertial sensors-based Mocap is more expensive, which impedes its wide applications. A full body inertial sensor-based Mocap system which includes 17 inertial units is shown in Fig. 1.6.

(a) (b)

Figure 1.6: (a) Inertial sensors-based Mocap system [18], (b) Real inertial units from Xsens [19].

1.3 Research Objectives and Challenges

Due to the passionate requirement from all kinds of applications of human motion analysis, our general research goal is to improve the performance of the markerless Mocap system either from monocular color camera or from a single depth sensor. To achieve this general goal, we mainly have three specific objectives which will be

presented from three perspectives, i.e., human motion modeling, body shape representation and efficient articulated pose estimation.

1.3.1 Human Motion Modeling

In video-based posture estimation, due to the high-dimensionality and variability of the motion data as well as the ambiguity from 2D imaging, it is challenging to reconstruct the optimal 3D posture in a high dimensional parametric space. Most methods rely on a prior motion model learned from training data to constrain the search in the solution space. In this research, one of our goals is to learn a powerful and representative human gait prior model which can support a more accurate walking motion estimation for different individuals and different motion types.

The 3D kinematical motion data is parameterized by the Euler angles or 3D positions of a set of body joints with the global 3D translation at the hip joint. For example, in our human motion modeling, gait kinematics of a body configuration is represented by a 59 dimensional vector (including 18 joints, each of which has 1-3 DOFs, and a 3D global translation at the hip joint). Obviously, there are high redundancy, high complexity and non-linearity in these high dimensional kinematical motion data. Therefore, the non-linear dimension reduction (NLDR) technique is necessary for dealing with the high dimensional kinematical data to achieve a compact representation as a statistical prior. Hence, we have the first research objective.

Objective 1: To learn a probabilistic non-linear low dimensional human gait model which can represent the walking motion from different subjects effectively and accurately.

To deal with more walking styles not only from different individuals, but also with different strides, our second goal is to explore and exploit a new latent structure for more complex gait modeling, while no additional training data are used.

Objective 2: To enrich the capability of motion representation of the

gait model by using a multi-layer structure in the latent space.

CMU Mocap dataset [34] is a standard human motion library. One challenge is that the human walking styles are very limited in CMU dataset and the model learned from this library is not diversified to represent various walking styles in reality. Another challenge is that the non-linear dimensional reduction-based learning process is computationally expensive and cannot be scaled up to a large scale training dataset. To this end, we first aim to diversify the walking styles artificially using the original limited CMU motion library. Second, we should have a new fast learning algorithm where a multi-layer low dimensional structure is designed to handle large-scale enriched training dataset. Then, we should employ an effective validation methods to evaluate the multi-layer model and compared with the single-layer model.

Once we have an effective motion model to constrain and refine the motion data from Mocap system, we can use it as a prior to obtain more accurate and robust motion data from video-based Mocap system. This research can lead to a practical and low-cost gait analysis technology for many real-world applications where traditional motion capture may be challenging through existing technology, such as in the hospital, nursing home or outdoor environment.

1.3.2 Articulated Shape Representation

A subject-specific body shape model is critical for accurate pose estimation. One challenge in the generative-based pose estimation is how to represent the body shape which not only can capture shape variability accurately, but is able to facilitate the data matching efficiently. To this end, our another goal in this research is to develop a simple yet effective parametric shape representation which can support efficient articulated pose estimation. Also, this general shape representation should be easily deformed to match a subject-specific body shape.

Objective 3: To represent the articulated body shape model using a set

of parametric models and to specify the shape model for different subjects.

1.3.3 Efficient Articulated Pose Estimation

Although the depth sensor has significantly simplify the task of human pose estimation, there still exists some challenges. First, most of the approaches rely on a large scale training dataset for retrieval or training a detector or predict model. In these methods, the pose estimation results are largely determined by the quantity of training data. However, collecting large scale training dataset is high-cost and time consuming and it is not available in many practical applications. Meanwhile, some other requirements, i.e., time-consuming training process, expensive hardware and complicated raw training data pre-processing hinder the development of this group methods. The second challenge is the computational complexity is still very high, due to the involved detailed mesh model and the inefficient energy function that is hard to be optimized. Most methods have to employ the GPU acceleration to achieve the real-time performance, which limits their applications on some mobile devices.

In this research, one of our goals on the depth sensor-based human pose estimation is to develop a fast and accurate Mocap system using a single depth sensor. Different with other state-of-the-arts methods, we aim to achieve the comparable accuracy and efficiency without any helps from database, detailed mesh model and GPU acceleration.

Objective 4: To develop an efficient and accurate articulated pose tracking from a single depth sensor by using a general parametric shape model.

There are several key challenges in this task. First, how to define an advanced energy function to support the efficient parameters estimation. Second, how to represent the transformation between body segments to construct the articulated skeleton. Third, how to augment the basic energy function with more constrains for the system accuracy and robustness. These questions guide our research in the pose estimation

parts and will be answered in the following Chapter 5, 6, 7.

1.4 Our Approaches

1.4.1 JGPM for Human Gait Modeling

Various non-linear dimension reduction (NLDR) approaches have been proposed for motion modeling, like GPLVM, GPDM and LL-GPDM. In [23], a two-step learning process was proposed to learn a torus-like low dimensional structure as a prior to constrain the human motion. In this research, inspired by the original JGPM, we introduce a new JGPM learning algorithm that is able to jointly optimize the gait and pose variables simultaneously, leading to a much compact parameter set and a straightforward procedure. Then, in order to compare our motion model with other state-of-the-art algorithms, we employ a validation technique and test all methods in terms of two motion analysis tasks, i.e., interpolation (to explain a unknown motion data) and filtering (to filter noisy Mocap data).

1.4.2 Multi-layer JGPM for Human Gait Modeling

To further enhance the representative capability of JGPM, we propose a new multi-layer JGPM that is capable of dealing with a variety of walking styles and various strides. Also, we can learn the model efficiently only from limited training data. Two new ideas are proposed. The first one is *training data diversification* that creates a series of simulated training gaits with different strides from a limited training dataset. This idea is inspired by several bio-mechanical experiments [35, 36, 37, 38], which reported that the human gait is left-right symmetrical and there exists certain proportional relation between limbs swinging to keep energy efficiency. The second one is *topology-aware local learning* that extends the stochastic gradient descent algorithm in [39] by only involving local neighbors according to the topology prior for model learning. Furthermore, we discuss two topological priors for coupling the pose

and gait manifolds in the latent space, cylindrical and toroidal, to examine their effectiveness and suitability for human motion modeling.

1.4.3 Articulated Shape Representation via Gaussian Kernels

To facilitate the template matching efficiently and represent the body shape in a simpler way, we first embed an articulated skeleton into a collection of multivariate Gaussian kernels where quaternion-based 3D rotations are involved to represent the transformation between two body segments along the skeleton. Then, based on our generalized Gaussian kernel correlation in Chapter 5, a segment-scaled articulated Gaussian kernel correlation is proposed to balance the effect of each segment in the articulated structure. Using the segment-scaled articulated Gaussian kernel correlation as an energy function, we propose an effective and efficient subject-specific shape modeling method, where a LLE-based topology constraint is involved as a regularization term. With this subject-specific shape modeling algorithm, the motion capture system could achieve more accurate pose estimation results for better motion analysis.

1.4.4 Articulated Pose Tracking

We formulate the articulated pose estimation problem as an optimization problem by defining a continuous and differentiable energy function based on our new generalized Gaussian kernel correlation which is a similarity metric. Therefore, the pose parameters (joint rotations) can be estimated by maximizing the similarity between the template and an observation. Due to the continuity and differentiability, our objective function for the articulated pose tracking can be efficiently optimized by gradient-based optimizers. In this research, we also augment the objective function for pose tracking with three additional constraints, i.e., visibility term, intersection penalty term and continuity term. Their derivatives have been explicitly derived to implement the fast gradient-based optimization. Moreover, we develop a failure

detection and recovery strategy to enhance the pose tracking with more robustness.

1.5 Our Contributions

In this research, we have two contributions on human gait modeling.

- First, different with torus-like JGPM, we present a new JGPM learning algorithm that is able to jointly optimize four variables simultaneously and the latent space is learned well in one-step process, leading to a much compact parameter set while sustaining a comparable performance with torus-like JGPM.

- Second, we propose a multi-layer JGPM to enhance the capability of motion representation, especially for the walking motion with wide motion ranges. To overcome the limitation of GPLVM-based learning for large-scale training dataset, we develop a neighborhood-based local learning strategy to handle huge training data which include all kinds of walking styles. The experiment results demonstrate the rationality and superiority of our proposed algorithm. This research has great potential in the applications of markerless motion capture system for gait analysis, motion tracking as well as character animation.

There are mainly three contributions on articulated pose estimation and shape representation.

- First, we extend the Gaussian kernel correlation function from the univariate case to the multivariate one in n dimensional space, along with a unified and differentiable similarity measure between any sum of univariate Gaussian kernels (SoG) and sum of multivariate Gaussian kernels (GSoG) combinations.

- Second, we present an articulated kernel correlation function for shape modeling and pose estimation where the tree-structured template is represented by a few multivariate Gaussian kernels along with a skeleton controlled by quaternion-based rotations.

- Third, we propose an efficient and robust sequential pose tracking algorithm by introducing three constraints (visibility, continuity and self-intersection) which is successfully applied to pose estimation of both body and hand from a single depth sensor.

1.6 Outline

This dissertation is organized as shown in Fig. 1.7. The summary of each chapter is briefly presented as follows.

- In Chapter 1, the motivation and significance of this research are presented.
- In Chapter 2, currently available works and methods for motion modeling and articulated pose estimation from a single depth sensor are reviewed and categorized.
- In Chapter 3, we propose a new joint gait pose manifold (JGPM) learning method and compare with other state-of-the-art algorithms.
- In Chapter 4, we provide the details of our multi-layer JGPM learning method and give the experimental results.
- In Chapter 5, we derive a generalized Gaussian kernel correlation function which extends the univariate Gaussian case to the multivariate one.
- In Chapter 6, we embed an articulated skeleton into a collection of Gaussian kernels to represent a shape model, and develop a subject-specific shape modeling method based on our proposed segment-scaled articulated Gaussian kernel correlation.
- In Chapter 7, we work on the articulated pose estimation from one depth sensor and our experimental results are shown.

- In chapter 8, we conclude our work and state our future work.

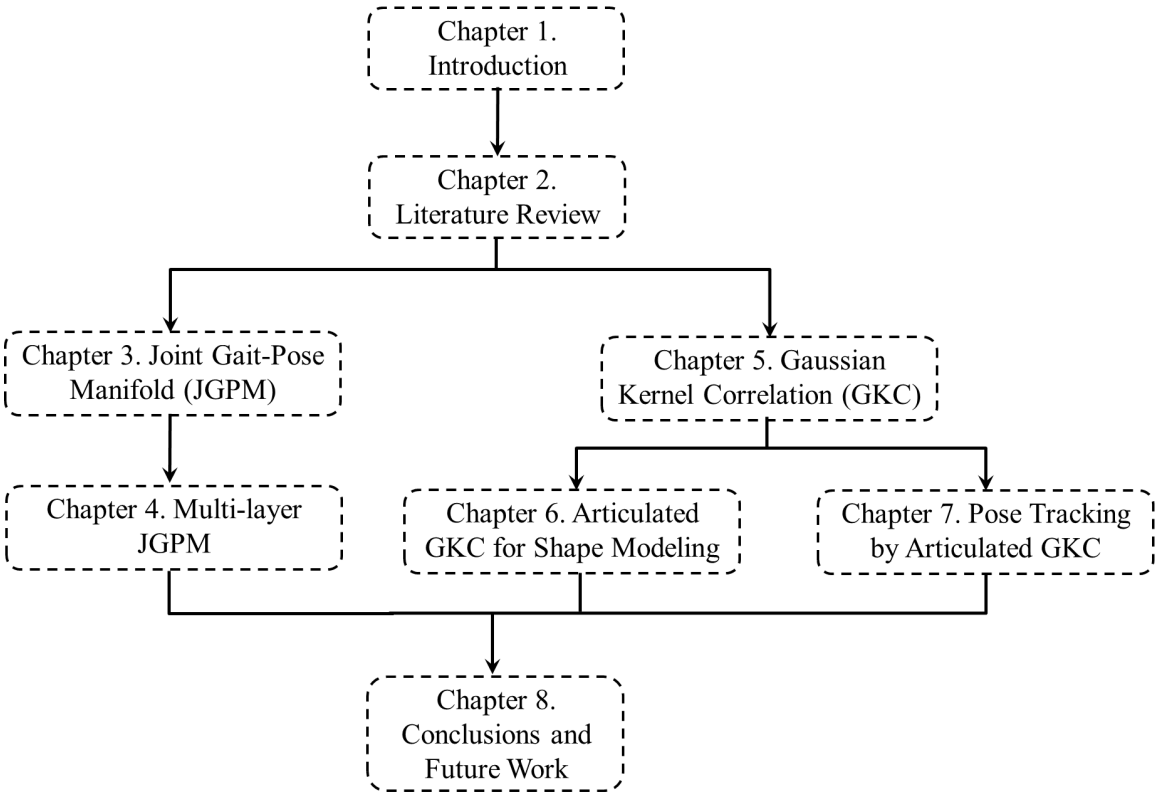


Figure 1.7: The outline of this dissertation.

CHAPTER 2

LITERATURE REVIEW

2.1 Overview of Human Motion Modeling

There exists a large body of research on human motion modeling. In early work, graphic models were used to represent the spatial and temporal priors of body parts [40, 41, 42]. An alternative is physics-based models which incorporate various kinematic/dynamic/physical constraints of body movements [43, 44, 45, 46, 47]. Recently, there are more and more non-linear dimensional reduction (NLDR)-based approaches, which try to explore the low dimensional intrinsic structure of human motion data. In this section, we provide a brief review of human motion modeling from a NLDR (or manifold learning) perspective with respect to three groups: *geometrically-inspired*, *latent variable model-based* and *hybrid algorithms*.

2.1.1 Geometrically-inspired Algorithms

The methods in this group seek to preserve the local geometrical neighborhood among high dimensional data in the low dimensional latent space through some unsupervised methods, such as Isometric Feature Mapping (Isomap) [48] and Local Linear Embedding (LLE) [22] that was applied successfully for human motion estimation without any initialization or prior constraints [49, 50]. However, Isomap and LLE provide neither a probability distribution over the latent space nor the mapping from the low dimensional latent space to the high dimensional data space. Given known topology, [51] developed a supervised topology preserving method for embedding data on a torus, where a separate mapping function (i.e., RBF-based mapping in [50]) is needed

by which the visual data can be associated with the kinematic data for video-based pose estimation.

2.1.2 Latent Variable Model-based Algorithms

The methods in this group apply the Gaussian Process (GP)-based approaches to provide a probability distribution over the latent space along with non-linear mapping function, such as GPLVM [52]. Many GPLVM variants have been proposed specially for human motion modeling, e.g., Back Constrained GPLVM (BC-GPLVM) [53], Gaussian Process Dynamic Model (GPDM) [54], Scaled GPLVM (S-GPLVM)[55] and Balanced GPDM (B-GPDM) [56]. Their relationship is shown in Fig. 2.1.

BC-GPLVM smoothes the trajectory of original GPLVM in the latent space by introducing a smooth mapping from the data space to the latent space. GPDM incorporate a temporal dynamical model in the latent space to smooth and regularize motion trajectories. Then, Balanced-GPDM improves the GPDM through balancing the influence of data reconstruction and the latent dynamics. Generally, these GPLVMs only involve one explicit factor in the latent space, i.e., pose. One excep-

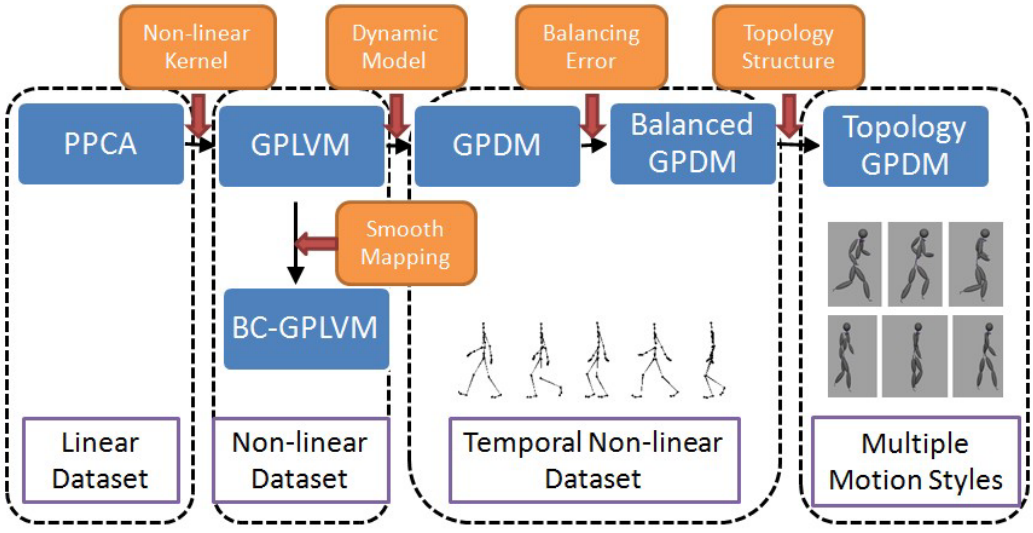


Figure 2.1: The road map of GPLVM and its variants.

tion is a multi-factor GP-based human motion model which was proposed in [57]. It incorporated multiple independent factors (i.e. identity, gait and state of motion) which are defined in different latent spaces for complex motion modeling.

2.1.3 Hybrid Algorithms

To preserve the geometrical neighborhood of latent structure, meanwhile to comply with the intrinsic data effect, the hybrid methods are developed. In [58], a topologically-constrained GPDM (LL-GPDM) was proposed to merge the pose manifolds from "walking" and "running" into the same cylindrical manifold structure by incorporating a LLE-based topology constraint into GPDM learning. The gait manifold was introduced in [59] to represent the variability of different gait styles which is learned by combining an idea similar to Isomap and non-linear tensor decomposition. The pose and gait manifold are assumed to be independent in [59]. To capture their coupling effect, a joint gait-pose manifold (JGPM) was proposed in [23, 60] by extending the LL-GPDM algorithm with a toroidal topological prior. It was shown that JGPM does improve video-based motion estimation results over the one in [59], and it also outperforms existing GP-based algorithms in terms of motion interpolation/reconstruction for normal human gaits. Still, JGPM may not be applicable to more complex gaits with various strides. Also, just like traditional GPLVM-based models, learning JGPM is computationally expensive and may not be scaled-up to a large training dataset with more subjects and various walking strides, which may limit its practical use.

2.1.4 Relationship with Our Research

Fig. 2.2 shows the taxonomy of manifold learning-based human motion modeling and their relationship with our research represented in red. In this research, inspired by the original JGPM in [23], we propose a new one-step JGPM learning algorithm that

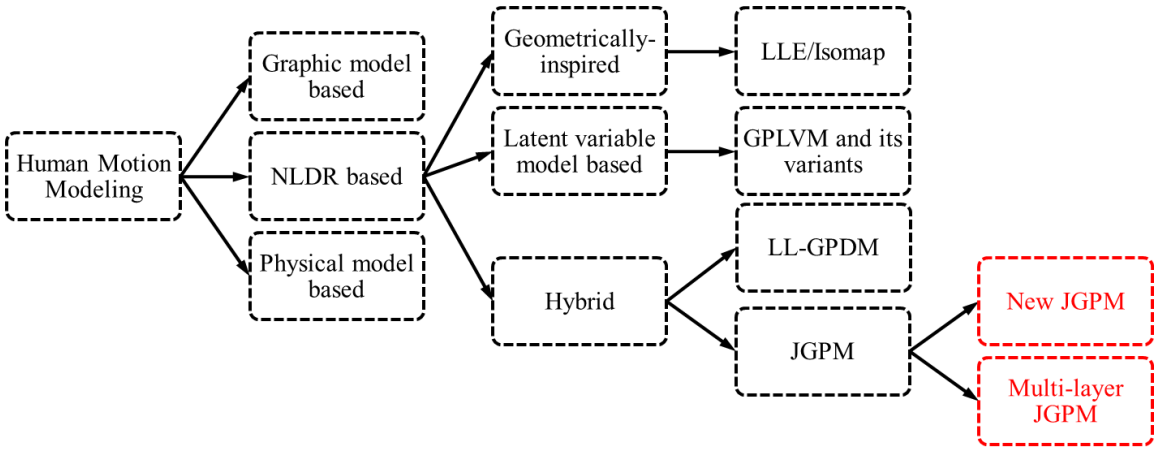


Figure 2.2: Taxonomy of manifold learning-based human motion modeling and their relationship with our research shown in red.

is able to jointly optimize the gait and pose variables simultaneously, leading to a much compact parameter set and a straightforward procedure. Also, to overcome the limitations of previous work in [23, 60], we develop a multi-layer JGPM, leading to a more representative model that is capable of dealing with a variety of walking styles with various strides. In our multi-layer JGPM, we introduce *training data diversification* to create more simulated training gaits and *topology-aware local learning* to make model learning more scalable and efficient. These two ideas could be applied to other problems where a general and powerful manifold model is desirable to deal with multiple latent factors of the data.

2.2 Overview of Articulated Pose Estimation

Recently, the launch of low-cost RGB-D sensors (e.g., Kinect) has further triggered a large amount of research on the articulated pose estimation due to the additional depth information and easy foreground/background segmentation. The existing algorithms can be roughly categorized into three groups, i.e., *discriminative*, *generative*

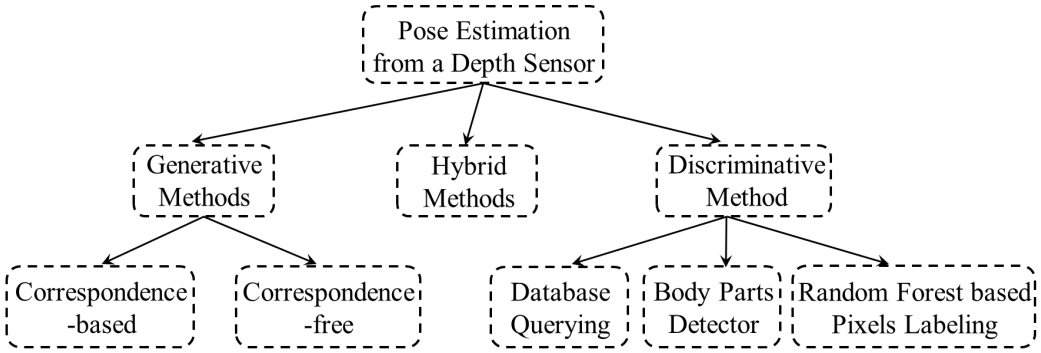


Figure 2.3: Taxonomy of articulated pose estimation from a single depth sensor.

and *hybrid* ones, as shown in Fig. 2.3. We compare most of the state-of-the-art algorithms reported so far in Table 2.1. Also, we will review the point set registration algorithms and the articulated shape representation, both of which are very fundamental and critical for the efficient pose estimation.

2.2.1 Discriminative Approaches

Discriminative approaches detect features in the depth image and then reconstruct a pose by either search in a database or directly predict the human body parts. For example, in [61], the body parts and their orientation were detected by identifying salient point of the human body. In [20], a random forest classifier was trained from a large scale dataset to label depth pixels into predefined human body parts, leading to a fast pose reconstruction. The procedure of this algorithm is shown in Fig. 2.4. Similarly, a regression forest based predictor was proposed in [62], which can predict the body joint positions directly. Also, similar discriminative approaches were proposed in [63, 64] for hand pose estimation.

While the discriminative methods can reconstruct the pose efficiently without initialization and they also can handle the large variation of body shape, the low accuracy of these methods limit their development in some applications, such as biomechanical, medical diagnose, etc. Additionally, most approaches in this group rely

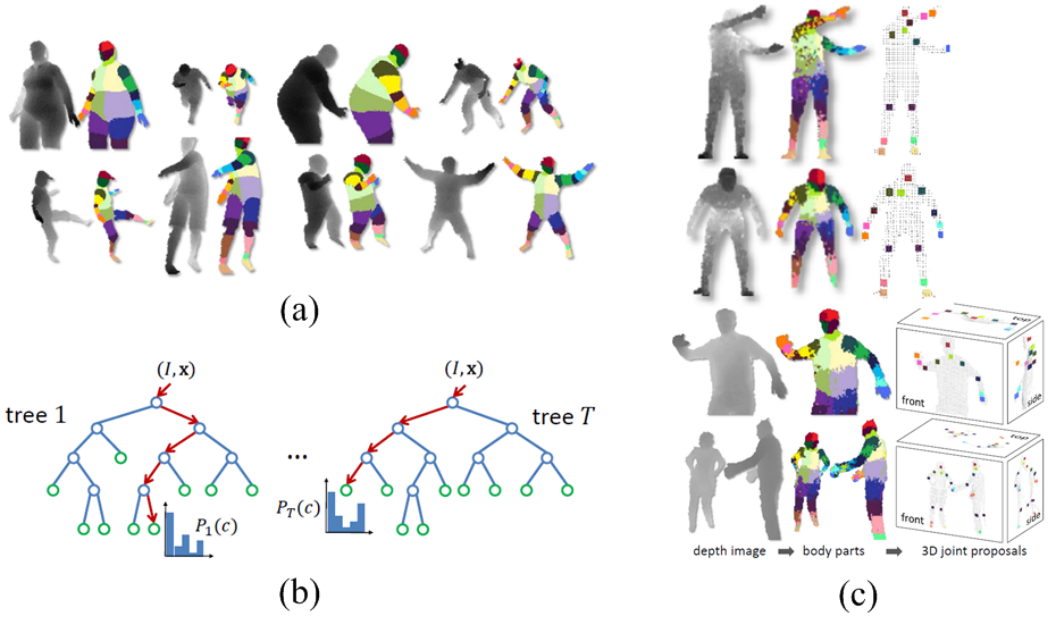


Figure 2.4: The algorithm procedure in [20]. (a) The illustration of training data. (b) The learning of random forest. (c) The inference of each depth pixel and the pose estimation results.

on a large scale dataset for training or retrieval, which is not available in many practical applications. Meanwhile, some other requirements, i.e., time-consuming training process, expensive hardware and complicated raw training data pre-processing hinder the development of this group methods.

2.2.2 Generative Approaches

Generative methods aim to estimate the parameters of a template model to best match the observed depth data. Most generative methods seek the explicit correspondence and then iteratively update the pose and correspondence, as show in Fig. 2.5 generated by [21]. Currently, the Iterative Closest Point (ICP) [65] and its variants, such as Articulated ICP [66], Non-rigid ICP [67] are the main methods for exploiting the correspondence. In [31], the author extended the ICP by modeling a "free space" constraint and proposed a tracking algorithm based on a Maximum

a Posterior (MAP) inference. The author in [68] used both depth and edge information to guide the tracker also within the ICP framework. Since estimating the correspondence is still challenging with noisy input and fast changing complex pose, these methods are prone to the local minima, leading to tracking failures. Without the explicit correspondence, a Gaussian Mixture Model (GMM)-based registration algorithm that is embedded with an articulated skeleton model was developed for human pose estimation using the Expectation-Maximization (EM) algorithm [28]. In [69], a discrepancy function was proposed for 3D articulated hand tracking which is optimized by a variant of Particle Swarm Optimization (PSO). This method was further extended in [70, 71].

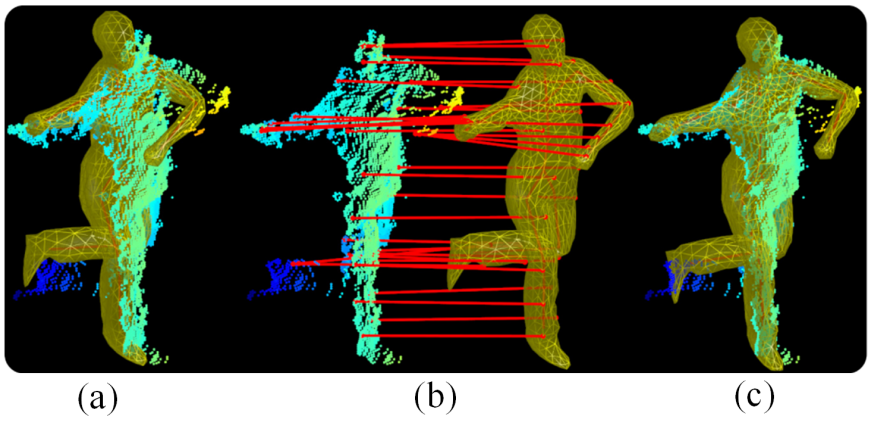


Figure 2.5: The pose estimation using generative approach in [21]. (a) The initial mesh template and observed point cloud, (b) the correspondence between the template and observation, (c) the estimated result.

While most of the generative approaches are capable to achieve higher accuracy compared with discriminative methods, they require a good initial pose to start the tracking; also they require a detailed mesh model or a geometrical body model constructed by point cloud. The computation complexity is very high in the generative methods, as they require hundreds of objective function evaluation during the alternative iteration between the optimization and correspondence seeking.

To capture human motion efficiently from multi-view 2D images, a shape model based on the sum of Gaussians (SoG) (i.e., the univariate SoG) was developed in [72]. This simple yet effective shape representation provides a (nearly) differentiable model-to-image similarity function, allowing fast pose estimation. SoG was also used in [73, 74, 29] for both human and hand pose estimation. In our early work [30], a generalized SoG model (GSoG) (i.e., the multivariate SoG) was proposed, where it encapsulated fewer anisotropic Gaussians for human shape modeling, and a similarity function between GSoG and SoG was defined in the 3D space. Sharing a similar spirit, a sum of anisotropic Gaussians (SAG) model was developed in [75] for hand pose estimation, where the similarity is measured by the projected overlap in 2D images. Both GSoG and SAG have improved the performance of pose estimation compared with the original SoG methods.

2.2.3 Hybrid Approaches

It is intuitive to take advantage of the complementary nature of the discriminative and generative approaches which involve querying or training data and useful data-driven detectors to assist the model-based optimization process. The hybrid methods have shown impressing results in [24, 21, 25, 76]. Ganapathi et al. [24] used body part detector [61] to benefit their ICP-based tracker. Baak et al. [21] detected five geodesic extrema to perform a Nearest-Neighbor search to locate an analogous pose as a competitor against the tracking result. Helton et al. [77] extended [21] by obtaining a personalized body shape for more accurate tracker. Similarly, Ye et al. [25] first looked up a database with the PCA of the normalized depth image to find a good initialization, and then it manipulated a deformable mesh model by seeking correspondence for accurate pose estimation. Wei et al. [78] combined the ICP-based tracking with the random forest classifier, the same algorithm with [20], to achieve tracking failure recovery. Most methods in this group are time-consuming and few of

them can perform real-time without the help of GPU. Here, we summary the features of most of the state-of-the-art approaches in Table 2.1.

2.2.4 Registration in Generative Approaches

Since the point set registration is the key technique in the generative approaches, we briefly review some registration methods which are highly related to our research. According to how the template and the target are matched, registration approaches can be classified into two major categories, i.e., correspondence-based and correspondence-free, which are shown in Fig. 2.6. The algorithms in the first category iteratively estimate the correspondences and the underlying transformation, such as the Iterative Closest Point (ICP) [65] and the Maximum Likelihood-based density estimation [79, 80, 81, 82]. When there are noise or outliers in the observation, the correspondence-based methods are prone to trap into poor local minima. To be robust to the noisy / outliers, the algorithms in the second group directly optimize an energy function without involving correspondences, including density alignment [83] and kernel correlation [84]. Different with the density alignment whose energy function is a discrepancy measure using L_2 distance, kernel correlation (KC) was first presented as a similarity measure in[85] and it was used for point set registration in [84], where both the template and the scene are modeled by kernels and their registration is achieved by maximizing a KC-based similarity measure. KC was also applied to the stereo vision-based modeling in [86].

When the kernel function is a Gaussian, there are two unique benefits for registration, i.e., robustness and efficient optimization. First, as stated in [83], Gaussian kernel correlation (GKC) in rigid registration is equivalent to the robust L_2 distance between two Gaussian mixture models (GMMs). Similarly, it was stated in [86] that GKC is equivalent to a distance measure between two data sets in the M-estimator [87]. Second, different from the Maximum Likelihood-based registra-

Table 2.1: Comparison of the state-of-the-art pose estimation algorithms

	Method	Database	Mesh	Accuracy	Efficiency	CPU/ GPU
Ganapathi et al. CVPR 2010[24]	Hybrid	Yes	Yes	100mm	6 fps	GPU
Baak et al. ICCV 2011[21]	Hybrid	Yes	Yes	62mm	60-100fps	CPU
Shotton et al. CVPR 2011[20]	Discriminative	Yes	Yes	NA	50 fps 200 fps	CPU GPU
Ye et al. ICCV 2011[25]	Hybrid	Yes	Yes	38mm	0.025 fps	CPU
Ganapathi et al. ECCV 2012[31]	Generative	No	Yes	NA	125 fps	CPU
Taylor et al. CVPR 2012[26]	Hybrid	Yes	Yes	37mm	120 fps	CPU
Wei et al. 2012 [78]	Hybrid	Yes	No	NA	30 fps	GPU
Kurmankh et al. 2013 [73]	Generative	No	No	NA	5 fps	CPU
Helton et al. 3DV 2013 [77]	Hybrid	Yes	Yes	60mm	NA	CPU
Ye et al. CVPR 2014[28]	Generative	No	Yes	34mm	30 fps	GPU
Ding et al. ISVC 2014[29]	Generative	No	No	56mm	5 fps	CPU
Ding et al. WACV 2015[30]	Generative	No	No	41mm	25 fps	CPU

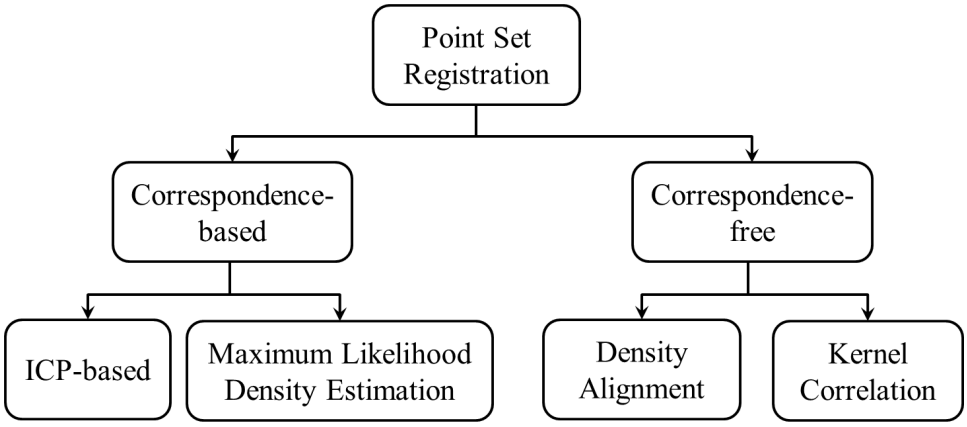


Figure 2.6: The classification of registration algorithms in generative approaches.

tion using Expectation-Maximization (EM) [80, 81, 82], the closed-form expression of GKC supports a direct gradient-based optimization which is more efficient and robust. However, existing GKC mainly considers the case of univariate (isotropic) Gaussian with two exceptions (to the best of our knowledge). First, SoG was extended to sum of anisotropic Gaussians (SAG) in [75] where the similarity function was evaluated in the projected 2D image space. Our previous work [30] studied anisotropic Gaussians in 3D space and derived a similarity measure between the template and target, represented by multivariate and univariate Gaussians, respectively. Both of works has their own limitation, which inspires our work in this dissertation.

2.2.5 Articulated Shape Representation

A good shape model not only captures shape variability accurately, but also facilitates the data matching efficiently. One of the most widely used shape models is the mesh surface which can depict the object precisely, but it usually involves a relatively high computational load and GPU-based implementation is often necessary for real-time processing [88, 28, 89]. Some other methods use a collection of geometric primitives, like spheres, cylinders or ellipsoids to render the object surface which is compared to the observed shape cues for matching [90, 91, 31, 78, 71]. On the other hand,

statistical parametric shape representations become more and more popular. Early work in [92] relied on simple 2D blobs. In [93], 230 implicit ellipsoidal metaballs constituted a body representation to simulate muscles and fat tissues. Moreover, a set of isotropic Gaussian components [72, 29, 74] and anisotropic ones [30, 75] are also employed to represent the humanoid body or hand. Compared with the mesh surface and geometric primitives representations, statistical parametric models are normally simpler with a lower computational load. It is worth noting that the geometric representation and parametric one are closely related but different on the way how the model is involved to compute the cost function during optimization. The category of human shape representation is shown in Fig. 2.7.

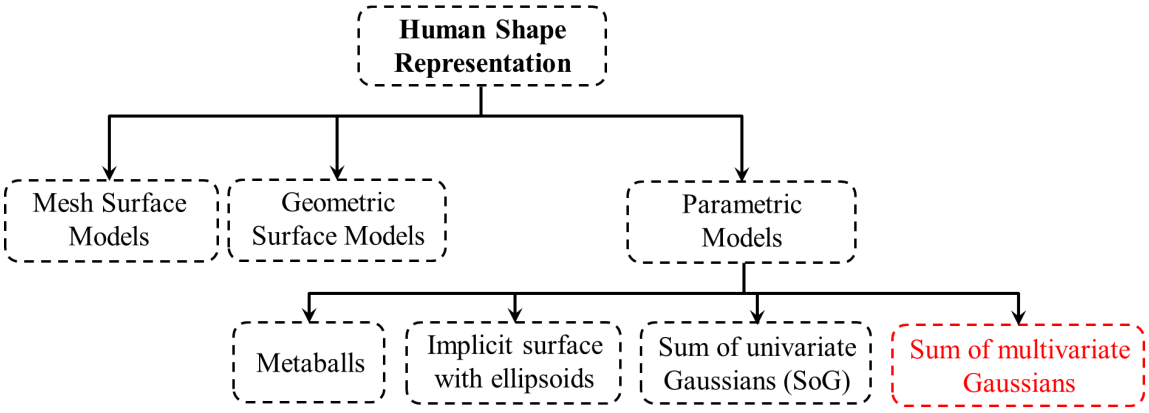


Figure 2.7: The category of human shape representation.

Although many 2D/3D shape representations have been proposed for rigid/non-rigid objects, to the best of our knowledge only a few shape representations are amenable to the articulated structure, due to the requirement of an underlying kinematic skeleton. In [28, 88], the detailed mesh model is able to be deformed by the twist-based transformation around the controlling joints for articulated pose estimation. While this method could achieve accurate results with a high computational load during the template matching, the complex blending between mesh and bones has to be considered. In [31], an articulated geometric representation was used to estimate the human pose by an improved ICP. As a parametric model, the SoG-based

model was straightforwardly embedded into a human skeleton in [72], and it was further used in [73, 29, 74] for the articulated human/hand pose estimation.

In this research, we embed a kinematical skeleton into Gaussian kernels and provide a more general parametric shape representation, which can be composed by a collection of univariate Gaussian kernels or multivariate ones or even the mixture of both. The general shape representation has some important advantages. First, it enables a continuous similarity measure with analytic gradients for efficient optimization. Second, its computational complexity is lower considering that fewer Gaussian kernels are involved. Third, it has better flexibility and adaptability for shape modeling by well approximating the elongated limb segments, blocky torso, and the rounded head. Last but not least, it allows a small variance along the depth direction to match the relatively flat point cloud data captured by a single depth sensor.

2.2.6 Relationship with Our Research

In this research, we provide a unified framework which generalizes all SoG-based approaches from the perspective of *Kernel Correlation*-based registration [84]. Specifically, we extend the Gaussian kernel correlation (GKC) from the univariate case to the multivariate one and derive a general similarity function between two collections of arbitrary Gaussian kernels, that is our unified framework is able to handle any pairwise comparison, including SoG↔SoG, SoG↔GSoG, GSoG↔GSoG, and even (SoG+GSoG)↔(SoG+GSoG). The last two new cases offer great flexibility and generality for articulated registration. We also embed a kinematic skeleton into the Gaussian kernels, leading to a simple yet effective shape representation and a tree-structured articulated GKC (AGKC) controlled by a group of quaternion-based rotations. Given the input point set represented by Gaussian kernels, pose parameters can be estimated by maximizing the AGKC between the shape template and the input data. Compared with the state-of-the-art generative approaches, our pose estima-

tion is simpler and more efficient with comparable accuracy due to the benefits from our AGKC-based objective function and fewer computational complexity. Compared with the discriminative and hybrid methods, our pose estimation does not use any database for training or querying, and our framework is general and applicable to the pose estimation of other structures with complex articulation, like hand or articulated mechanical parts.

CHAPTER 3

JOINT GAIT-POSE MANIFOLD LEARNING (JGPM)

3.1 Introduction

Human motion modeling is an active research topic in the field of computer vision and machine learning due to its wide applications, including video-based posture estimation for motion analysis, surveillance, and computer animation, robotics, etc. In video-based posture estimation, due to the high-dimensionality and variability of the motion data as well as the ambiguity from 2D imaging, it is challenging to reconstruct the optimal 3D posture in a high dimensional space. Comprehensive reviews on video-based human motion estimation and analysis are provided in previous surveys [94, 95, 96]. Most methods rely on a prior motion model learned from training data to constrain the search in the solution space [97, 98, 99, 100, 101]. In computer graphics animation, a good motion model is useful to synthesize various realistic poses of different motion types without specifying all of animation frames or degrees of freedom of a character [102, 103, 104] or to control the nonrigid deformation of skin and cloth [105]. In this chapter, we are interested in developing a probabilistic manifold-based motion modeling framework that is able to deal with a variety of walking styles from different individuals. We also adopt a set of metrics to compare the proposed motion model with existing ones in terms of the performance of motion modeling using a validation technique proposed in [106].

It is commonly believed that human motion data from a specific activity lie on a low dimensional manifold [107]. Recently, various nonlinear dimensionality reduction (NLDR) or manifold learning algorithms were proposed to learn a compact

low-dimensional motion prior to constrain the solution space for robust and accurate pose estimation. For example, Local Linear Embedding (LLE) [22] and Isometric Feature Mapping (Isomap) [48] were applied in [49], [50] for human motion modeling, where the local geometrical neighborhood among the high dimensional motion data is preserved in the low dimensional latent space. However, LLE and Isomap provide neither a probability distribution over the latent space nor a low-dimensional to high-dimensional mapping function. Some probabilistic NLDR methods, such as Gaussian Process Latent Variable Model (GPLVM) [52] and its variants [53, 54, 102, 56, 58] were developed for human motion modeling which provide a low-dimensional latent space along with a probabilistic mapping.

In NLDR-based human motion modeling, the term of *pose manifold* was often used to represent the sequential and cyclic pattern of human gait motion. The idea of *gait manifold* was introduced in [59] to represent the variability of different walking styles from multiple individuals, where dual gait generative models were proposed for motion modeling, one for visual data and one for kinematic data. The pose and gait manifolds are used independently to integrate two generative models for video-based motion estimation. To capture the coupling effect between pose and gait manifolds, a joint gait-pose manifold (JGPM) was proposed in [23], where a toroidal structure was employed to unify the pose and gait variables into one latent space and a two-step learning process was involved. Significant improvements were observed in [23] over that in [59], showing the benefit of joint modeling of pose and gait in the same manifold structure. Moreover, it was shown in [106] that JGPM shows promise compared with other GPLVM-based models in terms of the performance of motion modeling including motion interpolation, reconstruction, filtering and recognition.

In this research, inspired by the original JGPM in [23], we propose a new one-step JGPM learning algorithm that is able to jointly optimize the gait and pose variables simultaneously, leading to a much compact parameter set and a straightforward proce-

dure. Also, we employ a validation technique [106] to compare our proposed method with other state-of-the-art methods in terms of the motion modeling performance, which reveals that our method sustains a comparable performance with the original JGPM and still be superior to other existing GPLVM-based learning methods.

3.2 Preliminary

Given high dimensional observations, the key issue of human motion modeling is how to represent the high dimensional data by a general and compact low dimensional manifold. In this section, we introduce the preliminary background of our research, including Local Linear Embedding (LLE), Gaussian Process Latent Variable Model (GPLVM), Gaussian Process Dynamic Model (GPDM), Topologically-constrained GPDM (LL-GPDM) and the original JGPM.

3.2.1 Local Linear Embedding (LLE)

LLE [22] seek to maintain the local geometrical or linear proximity among the high dimensional data in the low dimensional manifold. One assumption of LLE is that each high dimensional data point and its neighbors lie on a locally linear patch on the low dimensional data manifold. In order to obtain the local geometry of these patches in low dimensional space, linear coefficients that reconstruct each high dimensional data point from its neighbors would be computed to characterize the local geometry. The work procedure could be concluded in three steps, which are illustrated in Fig. 3.1:

1. The K nearest neighbors $\eta_i = \{\mathbf{y_1}, \cdots, \mathbf{y_j}, \cdots, \mathbf{y_K}\}$ of each point $\mathbf{y}_i$ are computed in terms of Euclidean distance in the high dimensional space using $d_{ij} = \|\mathbf{y}_i - \mathbf{y}_j\|^2$;

2. The weight matrix $\mathbf{W}$ that best reconstruct each data point from its neighbors

is obtained by minimizing $\Phi(\mathbf{W}) = \sum_{i=1}^{N} \|\mathbf{y}_i - \sum_{j\in\eta_i} w_{ij}\mathbf{y}_j\|^2$, where w_{ij} is an element of $\mathbf{W}$;

3. Each latent points $\mathbf{x}_i$ that is best reconstructed by its neighbors according to the corresponding weights w_{ij} is computed by minimizing $\Phi(\mathbf{x}_1, \cdots, \mathbf{x}_N) = \sum_{i=1}^{N} \|\mathbf{x}_i - \sum_{j\in\eta_i} w_{ij}\mathbf{x}_j\|^2$;

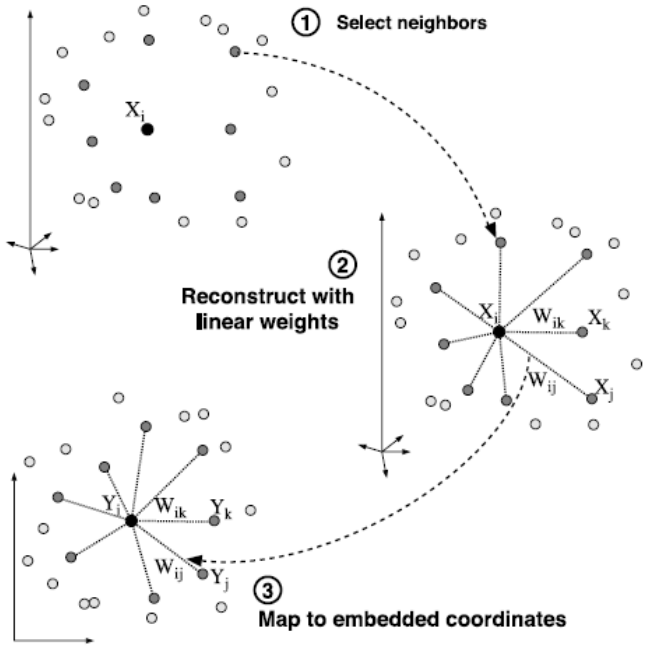


Figure 3.1: Algorithm flow of LLE [22].

In LLE, the two minimizations can be computed in a closed form. Particularly, computing the weight w_{ij} can be done by solving the following equation,

$$\sum_{k=1,k\neq j}^{K} C_{kj} w_{ij} = 1 \tag{3.1}$$

where k, j are the indexes of two neighbors of $\mathbf{y}_i$ and $C_{kj} = (\mathbf{y}_i - \mathbf{y}_k)^T(\mathbf{y}_i - \mathbf{y}_j)$. After the weight matrix is calculated, each latent point $\mathbf{x}_i$ can be obtained straightforwardly. In this way, LLE achieves the task of finding a topology through interconnections between points in the high dimensional space. However, the mapping between low dimensional points and high dimensional data can not be developed in the LLE.

3.2.2 GPLVM, GPDM and LL-GPDM

We first briefly review GPLVM, GPDM and LL-GPDM whose details can be found in [52, 54, 58]. GPLVM is a probabilistic manifold learning algorithm that can represent input data $\mathbf{Y}$ with a latent space $\mathbf{X}$ and can learn a low-dimensional to high-dimensional Gaussian process mapping. Let $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_N]^T$ ($\mathbf{y}_i \in \mathbb{R}^D$) represent the high dimensional data in which each row is a single training datum and $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_N]^T$ ($\mathbf{x}_i \in \mathbb{R}^d$) are corresponding latent points. GPLVM involves a likelihood function of the data given latent positions

$$p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\beta}) = \frac{1}{\sqrt{(2\pi)^{ND}|\mathbf{K}|}} \exp\left(-\frac{1}{2} tr\left(\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^T\right)\right), \tag{3.2}$$

where $\mathbf{K}$ is a $N \times N$ covariance matrix whose entries are defined by the kernel function, $\mathbf{K}(i, j) = k(\mathbf{x}_i, \mathbf{x}_j)$. The radial basis function (RBF) is often used as a kernel function. $\boldsymbol{\beta}$ denotes the kernel hyperparameters. GPLVM is learned by maximizing the likelihood in (3.2).

Considering the sequential nature of human motion data, GPDM [54] augments GPLVM by defining a GPLVM-based latent dynamical model $p(\mathbf{X}|\boldsymbol{\alpha})$ as

$$p(\mathbf{X}|\boldsymbol{\alpha}) = \frac{p(\mathbf{x}_1)}{\sqrt{(2\pi^{(N-1)D}|\mathbf{K}_X|)}} \exp\left(-\frac{1}{2} tr\left(\mathbf{K}_X^{-1}\mathbf{X}_{2:N}\mathbf{X}_{2:N}^T\right)\right), \tag{3.3}$$

where $\mathbf{X}_{2:N} = [\mathbf{x}_2, ..., \mathbf{x}_N]^T$, and $\mathbf{K}_X$ is the $(N-1)\times(N-1)$ kernel matrix constructed from $\mathbf{X}_{1:N-1} = [\mathbf{x}_1, ..., \mathbf{x}_{N-1}]^T$ and $\boldsymbol{\alpha}$ is the kernel hyperparameters for $\mathbf{K}_X$. GPDM incorporate a dynamic model as a prior into the latent space, leading to a smooth trajectory in the latent space which enables to interpolate or predict new motion data more accurately. It is worth noting that in both GPLVM and GPDM, the computation of inverse $\mathbf{K}^{-1}$ limits the scalability of this algorithm due to the fact that the computational complexity of $\mathbf{K}^{-1}$ grows cubically with the number of training data.

To model different motion activities ("walking" and "runing") in the same latent space, LL-GPDM [58] incorporates a LLE energy function $p(\mathbf{X}|\mathbf{W})$ in GPDM to

encourage a cylinder-shaped latent structure. Specifically, a predefined topological constraint is involved based on a neighborhood relationship learned via LLE. $\mathbf{W}$ is a weight matrix derived from the LLE, and LL-GPDM is learned by maximizing the *posterior* that is defined as

$$p(\mathbf{X}, \boldsymbol{\alpha}, \boldsymbol{\beta}|\mathbf{Y}, \mathbf{W}) \propto p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\beta})p(\mathbf{X}|\boldsymbol{\alpha})p(\boldsymbol{\alpha})p(\boldsymbol{\beta})p(\mathbf{X}|\mathbf{W}), \tag{3.4}$$

where $p(\mathbf{X}|\mathbf{W})$ is involved as a topology prior, $p(\boldsymbol{\alpha})$ and $p(\boldsymbol{\beta})$ are prior models for hyperparameters. LL-GPDM formulates a topological constraint and imposes it into GPDM framework. Although multiple motion types are embedded into one latent space, only one variable, i.e., *pose*, can be explicitly characterized. It is conceivable that an additional topology prior to order these pose manifolds may be helpful to reveal the underlying data structure across all walking/running cycles.

3.2.3 Original JGPM

Given a set of gait motion data from different individuals, the pose and gait are two essential variables for motion modeling. To unify the pose and gait variables into one latent structure, a joint gait-pose manifold (JGPM) was proposed in [23]. Due to the cyclic nature of the walking motion, the pose variable has a circular manifold. Since human gaits can not have huge dissimilarity, the author in [23] assume that the gait manifold is also a closed-loop structure. Consequently, a toroidal structure was employed as a topology constraint for the manifold learning, where a big horizontal circular shape in the torus represents a pose-specific gait manifold and a small vertical circular shape is a gait-specific pose manifold, as shown in Fig. 3.2.

In [23], three versions of JGPM with different levels of constraint were proposed, i.e., torus-based (JGPM-I), torus-constrained (JGPM-II) and torus-like (JGPM-III). JGPM-I employed an ideal and rigid torus structure and its learning process becomes a regression, where a two-way RBF mapping is involved. The manifold structure in JGPM-I cannot be adjusted because there is no consideration of the influence from

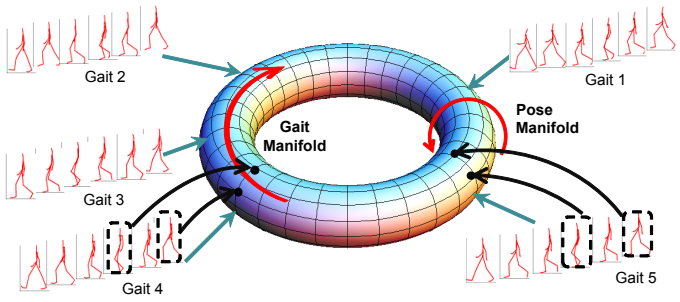


Figure 3.2: A toroidal structure for JGPM where the vertical and horizontal circles represent pose and gait manifolds, respectively [23].

training data. The JGPM-II still conform to an ideal torus. Initialized by a torus structure, JGPM-II optimized two angular variables which represent the pose and gait variable respectively so that an optimal latent space can be achieved. However, the JGPM-II still too rigid to reflect the underlying data structure.

JGPM-III encourages the manifold as a torus-like structure in 3D latent space through a two-step Gaussian Process-based learning, by which the JGPM can balance the effects from the rigid topology constraint and the intrinsic data-driven structure. The first step is to learn the pose manifold for each gait separately via GPDM, resulting in a set of local pose models. Then, these local pose models are aligned together to form a torus-like structure by GPLVM that optimizes a set of rigid transforms (including rotation and translation parameters) according to the gait manifold topology among all training gaits. Although the torus-like JGPM has more freedom to reflect the intrinsic data structure, the two-step GP learning process is computationally expensive and not straightforward. Moreover, the pose and gait variables are optimized separately in two latent spaces. In this research, we consider the coupling effect between pose and gait manifold and optimize these two manifolds in one latent space simultaneously.

3.3 Proposed JGPM

In this section, we propose a new JGPM learning algorithm by jointly optimizing four variables in the same latent space. Then, we employ a validation technique [106] to compare our propose JGPM with existing GP-based learning methods in terms of their capability of motion interpolation and filtering.

3.3.1 Toroidal Topology

In [23], a toroidal structure was used to learn JGPM. Specifically, a latent point on the torus surface belongs to a pose manifold for a specific gait (a vertical circle), meanwhile, it also belongs to a gait manifold at a specific pose (a horizontal circle). In the polar coordinate system, a torus can be parameterized by four variables $p, g \in [0, 2\pi)$ and R, r, which represent two angular variables *pose*, *gait*, as well as two radius values of the horizontal and vertical circles respectively. Hence, each latent point on the torus surface can be uniquely defined by $\mathbf{x}(p, g, R, r) = [t_x^{(p,g,R,r)}, t_y^{(p,g,R,r)}, t_z^{(p,g,R,r)}]^T$ as

$$\begin{aligned} t_x^{(p,g,R,r)} &= (R + r\cos(p))\cos(g), \\ t_y^{(p,g,R,r)} &= (R + r\cos(p))\sin(g), \\ t_z^{(p,g,R,r)} &= r\sin(p). \end{aligned} \tag{3.5}$$

The four torus-related variables are shown in Fig. 3.3, where each latent point $\mathbf{x}(p, g, R, r)$ corresponds to a high dimensional data point $\mathbf{y}_{(i,j)}$ which is the Euler angles of all body joints in the i_{th} pose and j_{th} gait. The topology of JGPM can be determined by the same method used in [23], and all latent points are initialized to be uniformly distributed on the torus surface along both angular variables. Next, we will integrate this toroidal topology constraint into a GPLVM-based energy function and develop a one step learning algorithm.

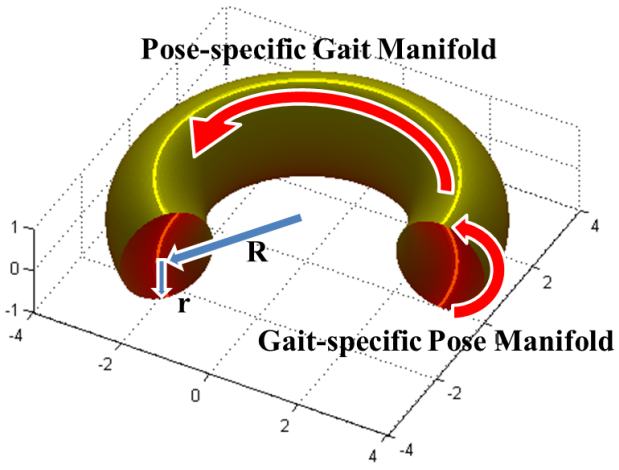


Figure 3.3: The illustration of *pose*, *gait*, R and r variables on JGPM.

3.3.2 One-step JGPM Learning

Extended from LL-GPDM [58], we develop a one-step learning algorithm for JGPM, where both gait and pose variables are involved. With the help of the constructed toroidal topology above and LLE, we can incorporate a specific prior into a GPLVM-based learning framework. Different from the original LLE, where the local linear neighborhood of high dimensional data was preserved in the low dimensional manifold, we aim to maintain the neighborhood of a specific low dimensional structure so that the manifold could resemble our prior. To this end, instead of finding the K nearest neighbors in the high dimensional data, we first define a set of adjacent points $\eta_i = \{\mathbf{x}_1, \cdots, \mathbf{x}_j\}$ for each latent point $\mathbf{x}_i$ on the torus. In this work, we select 10 nearest neighbors for each latent point as shown in Fig. 3.4. Then, to apply the toroidal topology, we construct the covariance matrix in LLE based on the prior structure and corresponding neighboring relationship. Given a latent point $\mathbf{x}_i = [t_x(i), t_y(i), t_z(i)]^T$ and its any two neighbors $\mathbf{x}_j, \mathbf{x}_k$, the corresponding covariance matrix element $C_i(j,k)$

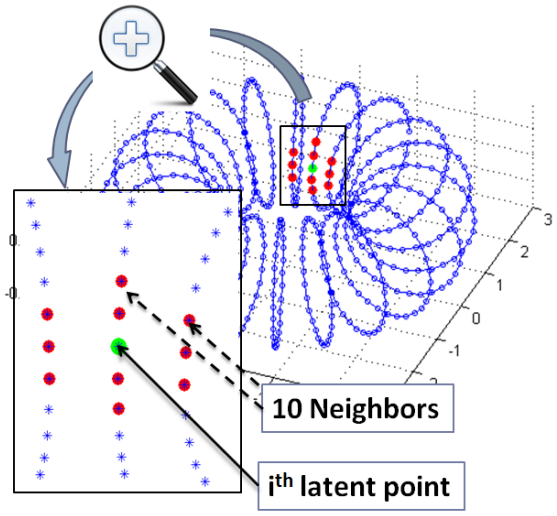


Figure 3.4: 10 Nearest Neighbors of each point on the surface of torus.

in each dimension is specified as

$$\begin{aligned} C_i^x(j,k) &= (t_x(i) - t_x(j))^T (t_x(i) - t_x(k)), \\ C_i^y(j,k) &= (t_y(i) - t_y(j))^T (t_y(i) - t_y(k)), \\ C_i^z(j,k) &= (t_z(i) - t_z(j))^T (t_z(i) - t_z(k)), \end{aligned} \tag{3.6}$$

Then the weight matrix $\mathbf{W}$ in each dimension can be computed by solving the following equations

$$\begin{aligned} \sum_{k=1,k\neq j}^{K} C_i^x(j,k) w_{ij}^x &= 1, \\ \sum_{k=1,k\neq j}^{K} C_i^y(j,k) w_{ij}^y &= 1, \\ \sum_{k=1,k\neq j}^{K} C_i^z(j,k) w_{ij}^z &= 1, \end{aligned} \tag{3.7}$$

where $C_i^x(i,j), C_i^y(i,j), C_i^z(i,j)$ are defined in (3.6). Note that the weights $\mathbf{W}$ should be different for each dimension according to their corresponding covariances. Given the weight matrix $\mathbf{W}$, we have the LLE energy function $p(\mathbf{X}|\mathbf{W})$ as

$$p(\mathbf{X}|\mathbf{W}) = \frac{1}{Z} \exp\{-\frac{1}{\sigma^2} \sum_{i=1}^{N} \|\mathbf{x}_i - \sum_{\mathbf{x}_j \in \eta_i} w_{ij} \mathbf{x}_j\|^2\}, \tag{3.8}$$

where $\mathbf{x}_i$ represents the i_{th} latent point, η_i is the collection of all neighbors of $\mathbf{x}_i$, w_{ij} is an element of the weight matrix $\mathbf{W}$, σ^2 is a scaling term and Z is a normalization term.

During the learning process, $p(\mathbf{X}|\mathbf{W})$ is defined as the topology prior that encourages latent points distributed as a toroidal structure. In other words, $p(\mathbf{X}|\mathbf{W})$ is larger when the latent points are closer to their prior distribution specified in $\mathbf{W}$. Then the learning process is to maximize the following posterior probability in terms of $\Pi = \{p_1, g_1, R_1, r_1\}, \cdots, \{p_N, g_N, R_N, r_N\}$, which represent the parameters of all the latent points,

$$\hat{\Pi} = \arg\max_{\Pi} p(\mathbf{Y}|\mathbf{X}(\Pi), \boldsymbol{\beta}) p(\mathbf{X}(\Pi)|\boldsymbol{\alpha}) p(\boldsymbol{\alpha}) p(\boldsymbol{\beta}) p(\mathbf{X}(\Pi)|\mathbf{W}) \tag{3.9}$$

where the first four terms are defined in GPDM, i.e., $p(\mathbf{Y}|\mathbf{X}(\Pi), \boldsymbol{\beta})$ is the likelihood function, $p(\mathbf{X}(\Pi)|\boldsymbol{\alpha})$ is the dynamic prior, $p(\boldsymbol{\alpha})$ and $p(\boldsymbol{\beta})$ are the hyperparameters for prior models.

We use the scaled conjugated gradient (SCG) optimization method to optimize the variables and other hyperparameters. Using this one-step GPLVM-based learning, we can obtain a new structure-guided JGPM, where latent points do not exactly conform to the ideal torus so that it can balance the intrinsic data structure with the topology constraint, and which is similar to the torus-like JGPM in [23] with a much lower complexity and higher training efficiency.

3.3.3 GPLVM-based Motion Model Validation

Given a set of noise-free training data, a GPLVM-based motion model learns a latent space including a prior motion model and a mapping between latent points and the high dimensional motion data. The well-trained motion model has the capability to explain the high dimensional motion data or to interpolation new motion data from a latent point. For example, a noisy input can be "projected" into the noise-free

latent space to find the best explanation that leads to a filtered motion sequence. The better model we learn, the more accurate filtering it has. Therefore, all GPLVM algorithms can be evaluated in the same way to compare their capability of handling various motion analysis tasks, including *interpolation*, *reconstruction*, *filtering*, and *recognition*. Specifically, *interpolation* is to synthesize a new motion sequence for an unknown subject (not from the training subjects); *reconstruction* is to recover full-body motion from partial one; *filtering* is to denoise noisy motion data from either a new unknown subject and a known training subject, and *recognition* is to recognize the identity from the underlying noisy motion data. Given a noisy input motion sequence $\{\mathbf{k}_1, \cdots, \mathbf{k}_T\}$, our goal is to estimate the latent points $\{\mathbf{x}_1, \cdots, \mathbf{x}_T\}$ and their corresponding kinematics $\{\mathbf{y}_1, \cdots, \mathbf{y}_T\}$ that maximizes the posterior probability defined as

$$p(\mathbf{y}_t, \mathbf{x}_t | \mathbf{k}_t) \propto p(\mathbf{k}_t | \mathbf{y}_t) p(\mathbf{y}_t | \mathbf{x}_t, \mathfrak{M}_{GP}) p(\mathbf{x}_t | \mathfrak{M}_{GP}), \tag{3.10}$$

where $\mathfrak{M}_{GP}$ is the learnt model. The first term $p(\mathbf{k}_t|\mathbf{y}_t)$ in equation (3.10) is the likelihood that measures the dissimilarity between testing and estimated kinematics using

$$p(\mathbf{k}_t | \mathbf{y}_t) = \exp(-\frac{f(\mathbf{k}_t, \mathbf{y}_t)}{\sigma^2}), \tag{3.11}$$

where σ^2 controls the sensitivity of evaluation and $f(\centerdot)$ is a dissimilarity measurement that indicates the degree of mis-match between two sets of motion data. According to [52, 54], the second term $p(\mathbf{y}_t|\mathbf{x}_t, \mathfrak{M}_{GP})$ represents the likelihood of the corresponding kinematics given a latent position, and it is defined as a Gaussian function of $\mathbf{x}_t$, that is, $\mathcal{N}(\mathbf{y}_t|\mu_Y(\mathbf{x}_t), \sigma_Y^2(\mathbf{x}_t))$ where,

$$\mu_Y(\mathbf{x}_t) = \mathbf{Y}^T \mathbf{K}_Y^{-1} \mathbf{k}_Y(\mathbf{x}_t), \tag{3.12}$$

$$\sigma_Y^2(\mathbf{x}_t) = k_Y(\mathbf{x}_t, \mathbf{x}_t) - \mathbf{k}_Y(\mathbf{x}_t)^T \mathbf{K}_Y^{-1} \mathbf{k}_Y(\mathbf{x}_t), \tag{3.13}$$

The third term $p(\mathbf{x}_t|\mathfrak{M}_{GP})$ is the prior probability of latent position $\mathbf{x}_t$ given the learnt latent space. This term could be a dynamic prior. For example, in the GPDM,

$p(\mathbf{x}_t|\mathfrak{M}_{GP})$ is defined as a Gaussian model $\mathcal{N}(\mathbf{x}_t|\mu_X(\mathbf{x}_{t-1}, \sigma_X^2(\mathbf{x}_{t-1})\mathbf{I})$ to characterize the dynamic model. More details of model validation techniques can be founded in [106]. This validation process is applied to all the GPLVM-based motion modeling algorithms in the following experiments.

3.4 Experimental Results

In this section, we report our experimental results in two parts. First, we compare five existing GPLVM-based algorithms, i.e., (1) GPLVM [52], (2)BC-GPLVM [53], (3) GPDM [54], (4) SB-GPDM [55, 56], (5) LL-GPDM [58] and the original JGPMs [23] with our proposed JGPM qualitatively through visualizing their latent space. Second, we perform motion model validation to compare them quantitatively in terms of a series motion analysis tasks. We implemented these GPLVM-based algorithms and the validation methods in Matlab with the reference code provided by Dr. Neil Lawrence.[1]

3.4.1 Experiment Setting

We chose 20 walking sequences (performed by 16 subjects) from the CMU Mocap Library [34] as the original training data, each of which contains 30 poses downsampled from about 130 frames in one walking cycle. Each pose is composed by 18 joints, including *lower/upper back, neck, left/right femur, tibia, foot, humerus, radius, wrist* and *thorax*, as well as *head*. The reason we chose 30 poses in one walking cycle is that involving more poses dramatically increases the computational complexity during the learning process, and more sparse training data could corrupt the smoothness in the latent space which is essential for the meaningful and realistic motion synthesis.

[1] http://staffwww.dcs.shef.ac.uk/people/N.Lawrence/software.html

3.4.2 Latent Space Comparison

We compare eight GPLVM-based models by showing the volumetric representation of their latent spaces in Fig. 3.5 where the color variation indicates the prediction confidence. Ideally speaking, a low dimensional latent space should reflect the intrinsic data structure in an intuitive and meaningful way, and a well-organized, smooth and compact manifold structure is usually preferred for human motion modeling. Since the GPLVM and BC-GPLVM were not originally designed for sequential data, their latent spaces were not very organized and smooth. Although GPDM and SB-GPDM integrate a dynamic latent model and show more meaningful motion trajectories, they still cannot to collectively represent multiple gaits in a unified way. LL-GPDM has a relatively well-defined cylinder-shaped manifold structure, where it only represent the pose manifold explicitly and treats the motion type variable implicitly. Among JGPMs, both JGPM-I and JGPM-II have an ideal torus while the former one involves a deterministic mapping relationship and the latter one is a probabilistic GP model. As we expected, the latent spaces of our new JGPM is much more organized and smoother. Deriving from ideal structures, the positions of latent points in the proposed JGPM are changed during the learning process to better reflect the high dimensional data and their neighboring relationship. Additionally, larger prediction confidence exists along the structure surface, implying better capability of motion interpolation and pose estimation. By comparing the new JGPM with the cylinder-shape latent structure of LL-GPDM, JGPM provides a better organized manifold structure to reflect both the commonality and variability of multiple gaits.

3.4.3 Quantitative Comparison

We quantitatively evaluate and compare nine advanced GPLVM-based algorithms in terms of the capability of unknown data representation (interpolation) and noisy data filtering through the model validation process mentioned in 3.3.3. We use the same

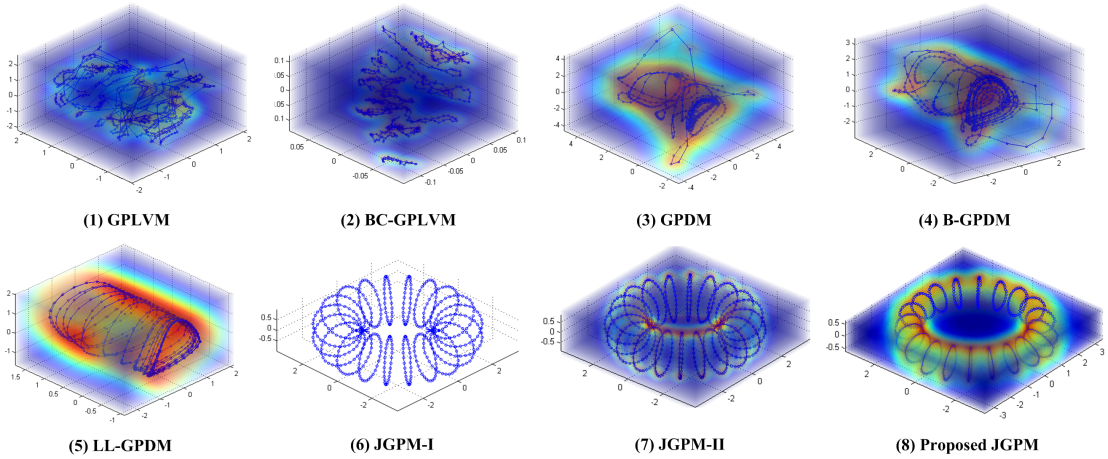


Figure 3.5: Volumetric visualization of prediction confidence variances in latent spaces; warmer colors, (i.e., red) depict lower variance.

kinematic training data as employed in the latent space comparison, including 20 different gaits from CMU Mocap Library. To collect the testing data, we selected 20 new motion sequences, among which ten are from known subjects (the same with training subjects, but from different walking cycles) and the other ten are from unknown subjects.

Motion Interpolation: To examine the data representation / synthesis capability, we chose ten unknown subjects as the test data and employed the validation method to interpolate new motion sequences from each of trained motion model. We computed the averaged 3D joint position errors (mm) between the estimated results and ground truth ones. The interpolation results are illustrated in Fig. 3.6. It is shown that the new JGPM provides the best performance that is comparable to the JGPM-III. BC-GPLVM, GPDM and B-GPDM are better than GPLVM due to the back-constraints or dynamic model involved. Because of the topology constraints, LL-GPDM further improves the results, but it is still surpassed by JGPM-II, JGPM-III and the proposed JGPM, showing the advantage of coupling the pose and gait variables into one manifold.

Motion Filtering: We used all 20 test sequences (10 unknown subjects and 10

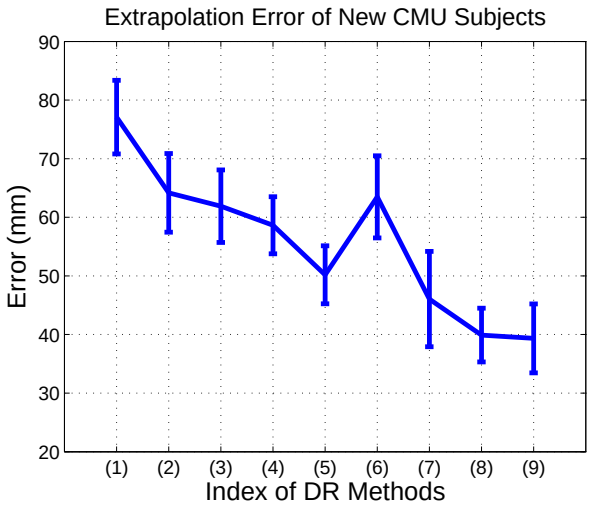


Figure 3.6: Comparison results of motion extrapolation, (1) GPLVM, (2) BC-GPLVM, (3) GPDM, (4) B-GPDM, (5) LL-GPDM, (6) JGPM-I, (7) JGPM-II, (8) JGPM-III, (9) the proposed JGPM.

known ones) to generate three types of noisy ones by adding different additive white Gaussian noise (AWGN), whose variances are 5%, 10% and 15% of frame-wise joint angle variation in a walking cycle. Then, we employ each trained model to filter the noisy motion data by the validation method. To verify the filtering effect, we visualize the results of our JGPM by stick man, as shown in Fig. 3.7, where we can observed that the estimated (filtered) results are closer to the original test skeleton (ground truth) than the noisy data.

Also, we show the filtering results of all trained models for known and unknown subjects respectively in Fig. 3.9 and 3.10, where we can observe that the proposed JGPM obtain the most competitive results for both known and unknown subjects.

3.5 Discussion

We have presented a new JGPM for human motion modeling that unifies the gait and pose variables into one latent structure. Compared with the original JGPM which is learned by a two-step learning process, a more straightforward one-step

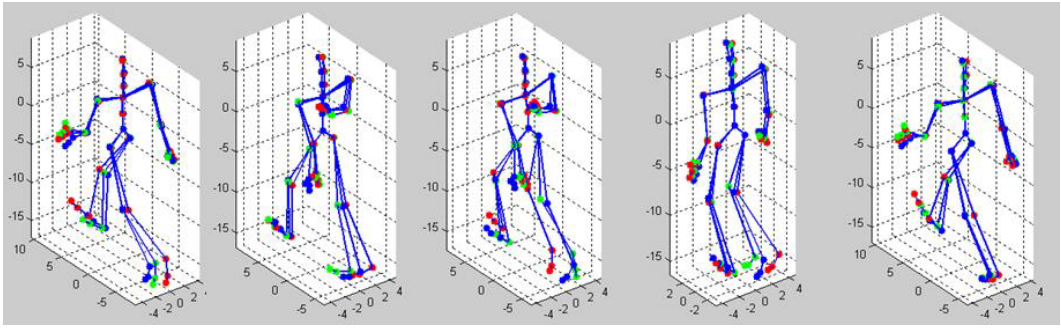


Figure 3.7: Visualization of the stick man for filtering experiment. The green points is the original test data; the red points represent the noisy data (noise level 10%) and the blue points is the filtering results.

GPLVM-based learning algorithm is developed in this chapter. Also, since less hyperparameters are involved, the computational complexity is considerably reduced, which makes it possible for large scale learning. Experimental results show that our proposed JGPM has the superior performance for motion interpolation and filtering compared with the existing GPLVM-based algorithms as well as the original JGPM-I and JGPM-II and it is comparable with JGPM-III in the numerical results.

One may doubt the validity the toroidal manifold prior for learning JGPM. Although the circular ordered nature of the pose manifold is easy to understand, that of the gait manifold is rather heuristic and driven by a few practical considerations. First, all human gaits are alike with, and a *closed* gait manifold is more plausible than an *open* one which implies some very dissimilar gaits. Second, a *closed* structure is preferred to provide a uniform neighborhood distribution and a continuous latent space which facilitate the learning and inference process. Third, the *circular* structure is a heuristic simplification that eases the learning and inference with the least number of free-parameters. Particularly, a "shortest-closed-path" technique was proposed in [23] to order all training gaits according their pair-wise similarities, leading to a *smooth* gait manifold where the similar gaits are clustered together while the dissimilar ones are separated. In summary, JGPM balances the toroidal mani-

fold prior and the intrinsic data structure, leading to a physically meaningful latent space as shown in Fig. 3.3 (a), where the manifold deviates from the ideal toroidal structure after learning. To further reflect its capability of reflecting the inherent data structure, Fig. 3.3 (b) compares the radii of all the pose manifolds with the dynamic variation of each individual gait along the gait manifold. It is observed that the radii of pose manifolds are highly related to the dynamic variation of corresponding gaits and they also show a smooth transition and the expected clustering effect along the gait manifold. we will compare the closed "toroidal" structure with an open "cylindrical" structure in Chapter 4.

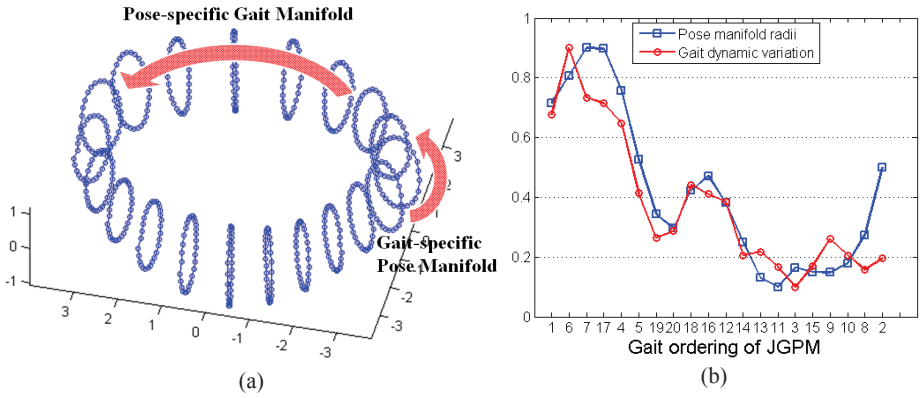


Figure 3.8: (a) The illustration of a trained JGPM in the 3D latent space. Each blue point represents one training sample. (b) The comparison of pose manifold radii and the corresponding gait dynamic variation.

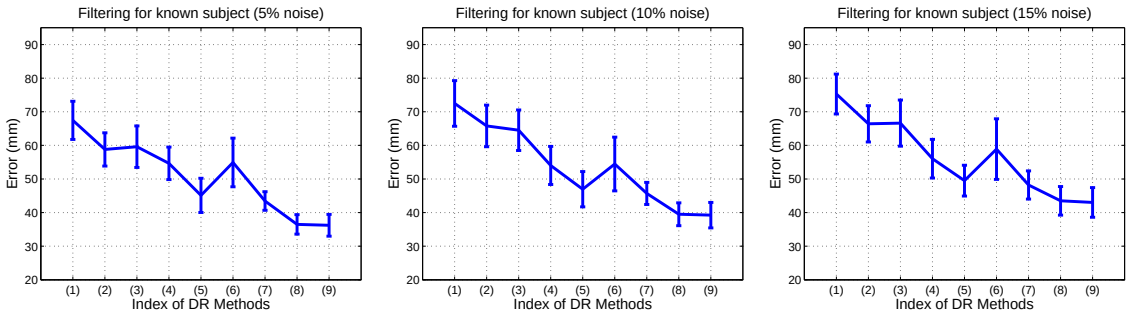


Figure 3.9: Comparison results of noisy filtering for known subjects, (1) GPLVM, (2) BC-GPLVM, (3) GPDM, (4) B-GPDM, (5) LL-GPDM, (6) JGPM-I, (7) JGPM-II, (8) JGPM-III, (9) the proposed JGPM.

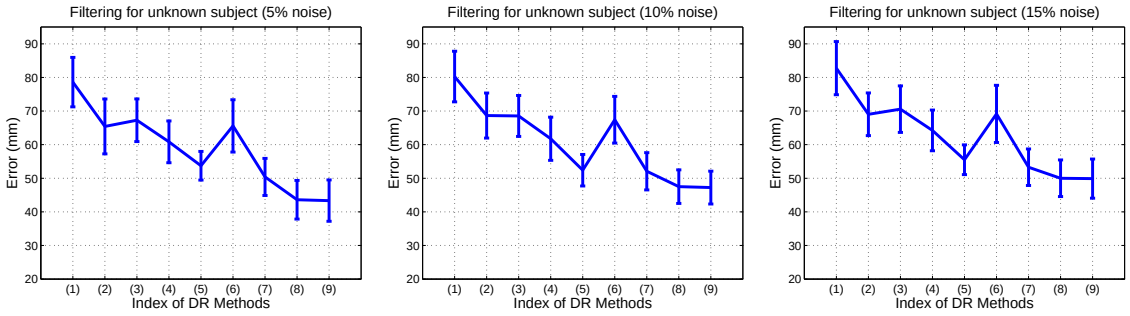


Figure 3.10: Comparison results of noisy filtering for unknown subjects, (1) GPLVM, (2) BC-GPLVM, (3) GPDM, (4) B-GPDM, (5) LL-GPDM, (6) JGPM-I, (7) JGPM-II, (8) JGPM-III, (9) the proposed JGPM.

CHAPTER 4

MULTI-LAYER JGPM

4.1 Introduction

In this chapter, we aim to develop a probabilistic manifold-based motion modeling framework that is able to deal with more walking styles not only from different individuals, but also with different strides while using the same training dataset with JGPM. Although it was shown in Chapter 3 that our new JGPM achieves promising results compared with other GPLVM-based models in terms of the performance on motion interpolation and noisy motion filtering, JGPM may not be applicable to complex gaits with different strides due to the limited training data. Also, just like traditional GPLVM algorithms, learning JGPM is computationally expensive and cannot be scaled up to a large training dataset. In this chapter, we propose a new multi-layer manifold model [108, 109] that is capable of dealing with a variety of walking styles and various strides. Also, we aim to learn the model efficiently from limited training data.

Two new ideas are proposed. The first one is *training data diversification* that creates a series of simulated training gaits with different strides from a limited training dataset. This idea is inspired by several bio-mechanical experiments [35, 36, 37, 38], which reported that the human gait is left-right symmetrical and there exists certain proportional relation between limbs swinging to keep energy efficiency. The second one is *topology-aware local learning* that extends the stochastic gradient descent algorithm in [39] by only involving local neighbors according to the topology prior for model learning. Furthermore, we discuss two topological priors for coupling the pose

and gait manifolds in the latent space, cylindrical and toroidal, to examine their effectiveness and suitability for human motion modeling. The experiments demonstrate that our proposed both multi-layer JGPMs have great flexibility and capability of representing a wide ranges of gaits with very different strides compared with the single layer JGPM and other GPLVM-based methods. Moreover, it is interesting to find that the toroidal prior is slightly better than the cylindrical one in our study. We believe it is mainly due to the fact that the closed nature of the toroidal structure supports a uniform neighborhood structure along the manifold which in turn facilitates learning and inference. It is worth noting that our motion modeling algorithm is not limited to human gait data only, but it could also be applied to other kinds of data, like hand-written digits and face expression.

4.2 Multi-layer JGPM

4.2.1 Motivation

Fig. 4.1 shows a latent space with two circular-shaped concentric manifolds learned by GPLVM from a dataset of two rotated hand-written digit series with different sizes. This example shows a simple two-layer structure in the latent space, which inspires us to introduce a multi-layer manifold structure for human motion modeling. It is easy to view Fig. 4.1 from the perspective of Principal Component Analysis (PCA). The two image subsets that have the same rotated digits at two different scales should have similar eigenvectors used to span the low dimensional space. The radius of the circular-shaped manifold is represented by the magnitude of the data projection on the first two eigenvectors, and it is proportional to the standard deviation of the high dimensional data. In this work, we will explore this *multi-layer manifold learning* idea in the context of complex human gait modeling, where our objective is to enhance the representativeness and diversity of the motion model.

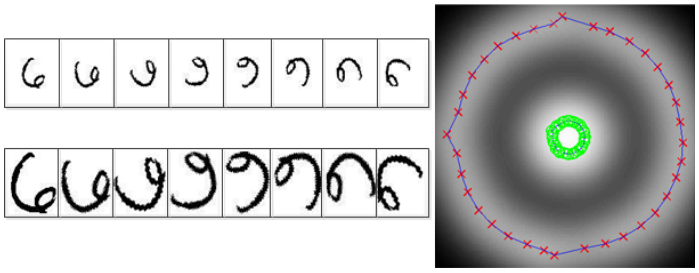


Figure 4.1: Two approximately circular manifold of the rotated digits dataset are learned by GPLVM in a 2D latent space. The inner and outer circular structure (green and red) represent the smaller and larger rotated digits, respectively.

4.2.2 Training Data Diversification

Intuitively, including more diverse walking styles as the training data is helpful to enhance the flexibility of general motion modeling. However, it may not be practical to collect a large motion dataset from many subjects, and it would be practically useful if we can *generate* more simulated motion data from a limited training set. Given a set of body joints defined by a skeleton model, human motion data are usually represented by the 3D positions or 3D Euler angles at each joint. Especially, the latter representation can directly reflect the motion range of each body segment during a gait. Inspired by some biomechanics evidences [35, 36, 37, 38], it is intriguing to use multiple scaling factors to diversify the training data by adjusting the standard deviation while maintaining the mean of Euler angles at each joint, by which a multi-layer manifold could be learned to represent a variety of walking styles with diverse motion ranges.

One major assumption behind this motion scaling idea is that a new gait can be approximated by a training gait by scaling the dynamic range of Euler angles at each joint. Although this assumption is worth further scrutiny, we will take this idea to diversify the original training data in order to learn a more flexible motion model. Let $\mathbf{y}_{u,v}^{(k)}$ represents a 3D Euler angle vector including three rotations, i.e., pitch, yaw

and roll, where u, v denote u^{th} pose in v^{th} gait sequence and k is the bone joint index. The new simulated motion data $\mathbf{y}^{(k)}_{u',v'}$ is generated by

$$\mathbf{y}^{(k)}_{u',v'} = \frac{1}{n}\sum_{u=1}^{n}\mathbf{y}^{(k)}_{u,v} + s \cdot \big(\mathbf{y}^{(k)}_{u,v} - \frac{1}{n}\sum_{u=1}^{n}\mathbf{y}^{(k)}_{u,v}\big), \tag{4.1}$$

where n is the number of poses in a gait sequence and s is the scaling factor. In practice, it was found that a scaling factor between (0.3-1.5) can lead to a realistic looking gait. Fig. 4.2 (top three rows) shows two motion scaling examples, where two scalars, 1.25 and 0.5, are used to create two scaled gait sequences, and the corresponding 2D latent spaces generated by Back Constraint GPLVM [53] and PCA are shown in Fig. 4.2 (a) and (b). The two latent spaces reveal some interesting relationship between the motion ranges and the radii of pose manifolds, i.e., a wider motion range results in a larger radius of the learned pose manifold, vice versa. As shown by latter experiments, this simple yet effective way can multiply a limited training dataset with more diversity and variability.

4.2.3 Multi-layer Structures

While the toroidal and cylindrical structures are two heuristic designed prior, they are simple yet physically meaningful, and helpful to organize the latent space, which makes the inference more accurate. Correspondingly, we introduce a three-layer toroidal and a three-layer cylindrical structure as shown in Fig. 4.3 (a) and (b) as new topology priors to initialize the multi-layer JGPMs. The outer layer represents the motion data which have a larger range (e.g. scalar 1.25); those with a smaller range (e.g. scalar 0.4) are embedded into the inner layer; the middle layer represents the original motion data. Hence, every initial point indexed by (p, g, s) on the toroidal

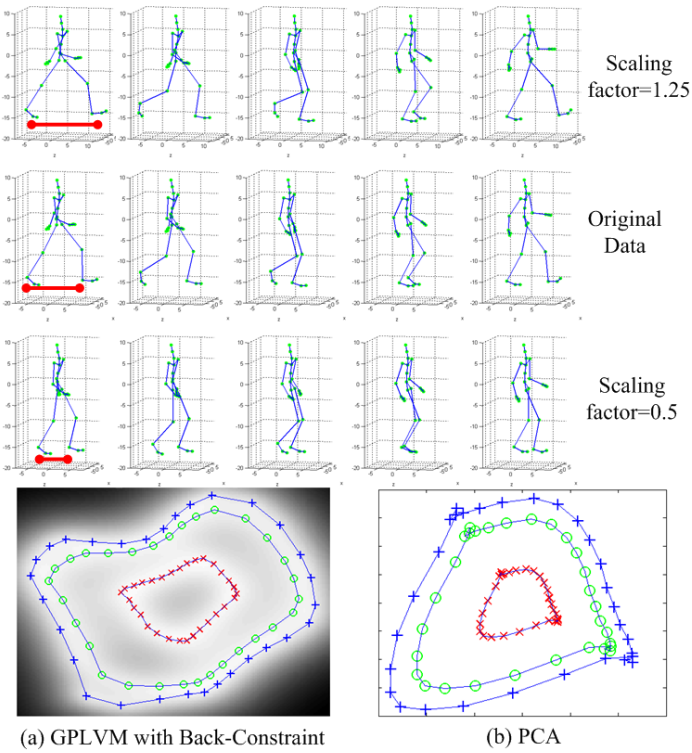


Figure 4.2: Illustration of the scaled motion and two latent spaces generated by GPLVM with back constraint and PCA respectively.

structure can be uniquely defined by a 3D coordinate $[t_x^{(p,g,s)}, t_y^{(p,g,s)}, t_z^{(p,g,s)}]^T$ as

$$\begin{aligned} t_x^{(p,g,s)} &= (R + r^{(s)}\cos(\alpha))\cos(\beta), \\ t_y^{(p,g,s)} &= (R + r^{(s)}\cos(\alpha))\sin(\beta), \\ t_z^{(p,g,s)} &= r^{(s)}\sin(\alpha), \end{aligned} \tag{4.2}$$

where p, g, s are the indexes of *pose*, *gait* and *scale*; α and β are two angular values corresponding pose p of gait g. R and $r^{(s)}(s = 1, 2, 3)$ are the radii of one horizontal (along the gait manifold) and three vertical circles (along three pose manifolds). In cylindrical structure, the gait manifold is a open-loop line structure. Similarly, every

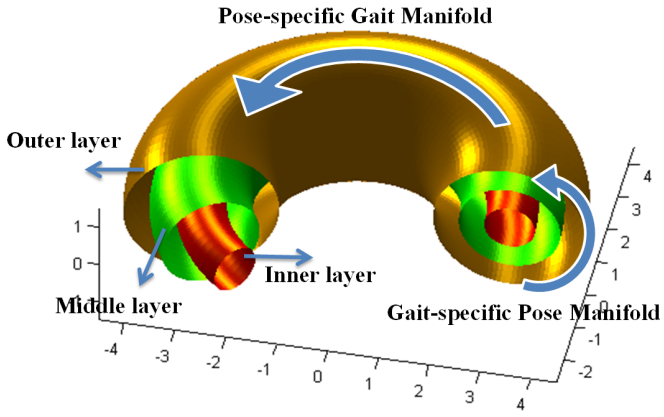
initial point is defined by a 3D coordinate $[t_x^{(p,g,s)}, t_y^{(p,g,s)}, t_z^{(p,g,s)}]^T$ as

$$\begin{aligned} t_x^{(p,g,s)} &= r^{(s)} \cos(\alpha), \\ t_y^{(p,g,s)} &= \delta \cdot g, \\ t_z^{(p,g,s)} &= r^{(s)} \sin(\alpha), \end{aligned} \tag{4.3}$$

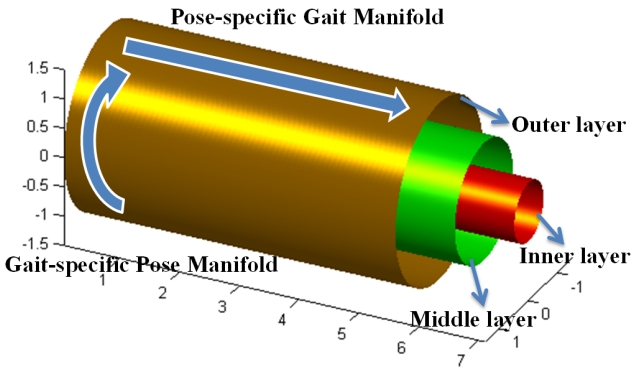
Different with toroidal structure, the coordinate $t_y^{(p,g,s)}$ is represented by δ (the interval of two adjacent gaits) multiplying the index of gait g. Empirically, the *interval* is defined as 0.5 for example considering the radius of middle layer circle is 1. For the convenience in the subsequent section, without special statement, we will not show the equations of cylindrical structure version separately due to its similarity with the toroidal structure version.

This three-layer structure will be used to initialize the multi-layer JGPM. The gait topology of each layer (the ordering relationship of all training gaits, i.e., g variable) is computed by classical traveling salesman problem in the close-loop toroidal structure or by shortest path problem in the open-loop cylindrical structure, respectively. We re-order the training gait according to the gait topology to make sure similar training gaits are close to each other, vice versa.

One thing worthy discussing is the comparison of toroidal and cylindrical structure, both of which capture the essential variables for the human motion modeling, i.e., pose and gait. The differences between the two latent structures are the pattern of gait variable. Toroidal structure utilities a close-loop structure considering the intrinsic similarity among all human gaits, that is none of two gaits are extremely different. On the other hand, the structure prior of gait manifold could not be limited to closed-loop. We further test the open-loop cylindrical structure and its corresponding cylindrical JGPM. Observed from our experimental results in Section 4.4, the toroidal JGPM achieves slightly better performance than the cylindrical JGPM. The main reason could be the absence of training gaits and the limited neighborhood



(a)



(b)

Figure 4.3: (a) A three-layer torodial structure as a topology prior. (b) A three-layer torodial structure as a topology prior.

configuration at the two ends of the cylindrical structure, which have some effects on the model learning. On the contrary, the toroidal structure does not confront the two limitations above due to its close-loop property. Additionally, the toroidal structure could be more suitable for the video-based motion estimation through sampling in the trained latent space.

4.2.4 LLE-based Topology Constraint

After constructing the multi-layer structures, we need to incorporate these specific topology priors into a GPLVM-based learning framework. It is worth mentioning first that the topology constraints method is suitable for both toroidal and cylindrical

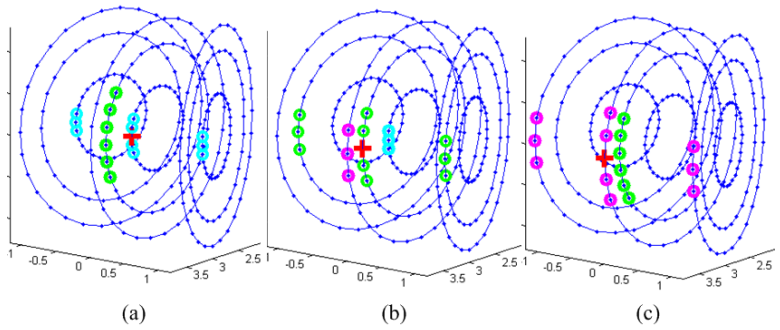


Figure 4.4: Neighborhood configurations in the topology constraint for a reference point (red cross) on (a) inner layer (b) middle layer (c) outer layer. Different colors mean the neighbors are from different layers.

structure. Different from the original LLE method, where the local neighborhood relationship of input data was preserved in the manifold, our aim is to maintain the neighborhood of a specific latent structure so that the learned manifold could resemble our topology prior. Therefore, instead of finding the K nearest neighbors in the data, we first define a set of adjacent points $\{\mathbf{x}_j\}_{j\in\eta_i}$ for each point $\mathbf{x}_i$, where η_i is the collection of all neighbors for the i^{th} point. To preserve both the topological structure within a layer and across layers, η_i should include some within-layer and cross-layer neighbors. Specifically, for a given point $\mathbf{x}_i$, $\eta_i = \{\phi^{(i)}_{1\cdots m}, \psi^{(i)}_{1\cdots n}\}$ which store the indexes of m within-layer and n cross-layer neighbors. The basic principle of neighbor selection is that we expect to have a stronger within-layer constraint than the cross-layer one, i.e., $m > n$. Fig. 4.4 shows an example of neighbors collection for one reference point.

In LLE, the definition of covariance $C_{jk} = (\mathbf{y}_i - \mathbf{y}_j)^T(\mathbf{y}_i - \mathbf{y}_k)$ with $j, k \in \eta_i$ is used to compute the wight matrix $\mathbf{W}$ in high dimensional space. To reflect the prior knowledge, i.e., the multi-layer toroidal/cylindrical topology, we specify a unique covariance matrix for each latent dimension using the coordinates of latent points in

(4.2):

$$
\begin{aligned}
C_{jk}^{x} &= \big(t_{x}^{(p_i,g_i,s_i)} - t_{x}^{(p_j,g_j,s_j)}\big)^{T}\big(t_{x}^{(p_i,g_i,s_i)} - t_{x}^{(p_k,g_k,s_k)}\big),\\
C_{jk}^{y} &= \big(t_{y}^{(p_i,g_i,s_i)} - t_{y}^{(p_j,g_j,s_j)}\big)^{T}\big(t_{y}^{(p_i,g_i,s_i)} - t_{y}^{(p_k,g_k,s_k)}\big),\\
C_{jk}^{z} &= \big(t_{z}^{(p_i,g_i,s_i)} - t_{z}^{(p_j,g_j,s_j)}\big)^{T}\big(t_{z}^{(p_i,g_i,s_i)} - t_{z}^{(p_k,g_k,s_k)}\big),
\end{aligned}
\tag{4.4}
$$

where $j,k \in \eta_i$. (p_i,g_i,s_i), (p_j,g_j,s_j), and (p_k,g_k,s_k) are indexes of $\mathbf{x}_i$, $\mathbf{x}_j$ and $\mathbf{x}_k$, respectively, from which we can find the 3D coordinates of three points according to (4.2). To compute $\mathbf{w}_i$ in each dimension, i.e., $\{\mathbf{w}_i^{(\tau)}|\tau \in (x,y,z)\}$, we solve the following equations:

$$
\begin{aligned}
\sum_{k} C_{jk}^{x} w_j^{x} &= 1,\\
\sum_{k} C_{jk}^{y} w_j^{y} &= 1,\\
\sum_{k} C_{jk}^{z} w_j^{z} &= 1,
\end{aligned}
\tag{4.5}
$$

where $C_{jk}^{x}, C_{jk}^{y}, C_{jk}^{z}$ are defined in (4.4), and then normalize the weight vector . Given the whole weight matrix $\mathbf{W}$, which is comprised by $\mathbf{w}_i^{(\tau)}$, where $i = 1,...,N$ and $\tau \in (x,y,z)$, the LLE energy function $p(\mathbf{X}|\mathbf{W})$ is defined as

$$
p(\mathbf{X}|\mathbf{W}) \propto \prod_{\tau\in(x,y,z)} \exp\{-\frac{1}{\sigma^2}\sum_{i=1}^{N}\|\mathbf{x}_i^{(\tau)} - \sum_{j\in\eta_i} w_{ij}^{(\tau)}\mathbf{x}_j^{(\tau)}\|^2\},
\tag{4.6}
$$

where $\mathbf{x}_i^{(\tau)}$ represents a coordinate of $\mathbf{x}_i$ along dimension τ, $w_{ij}^{(\tau)}$ is an element of $\mathbf{w}_i^{(\tau)}$ and σ represents a scaling term. Using the energy function above, we can incorporate the topology constraint into the LL-GPDM learning framework defined in (3.4) to encourage the manifold to resemble the topological prior.

4.3 Topology-aware Local Learning

Traditional GPLVM-based learning algorithms struggle to learn a model from a large-scale dataset, because the computation complexity grows cubically with the number

of training samples. Here we seek to a fast and effective GPLVM-based local learning algorithm for the diversified training data, which is termed as *Topology-aware Local Learning*. This *Topology-aware Local Learning* is general to various topology structures, including toroidal and cylindrical structure.

GPLVM is learned by maximizing the likelihood in (3.2), which is equivalent to minimize the negative log likelihood

$$\begin{aligned}\mathcal{L} &= -\ln p(\mathbf{Y}|\mathbf{X},\beta) \\ &= -\frac{DN}{2}\ln(2\pi) - \frac{D}{2}\ln|\mathbf{K}| - \frac{1}{2}tr\left(\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^T\right),\end{aligned} \tag{4.7}$$

To minimize $\mathcal{L}$, the gradient of $\mathcal{L}$ with respect to $\mathbf{X}$ is computed as

$$\frac{\partial \mathcal{L}}{\partial \mathbf{X}} = \frac{\partial \mathcal{L}}{\partial \mathbf{K}} \cdot \frac{\partial \mathbf{K}}{\partial \mathbf{X}} = -\left(\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^T\mathbf{K}^{-1} - D\mathbf{K}^{-1}\right) \cdot \frac{\partial \mathbf{K}}{\partial \mathbf{X}}, \tag{4.8}$$

where $\mathbf{K}$ is the $N \times N$ kernel matrix, where N is the number of training data. The computation complexity of $\mathbf{K}^{-1}$ is $O(N^3)$, which considerably limits the application of GPLVM for large-scale training dataset. The main idea of existing sparsification techniques [110, 111, 112, 113] is to reduce the dimensionality of the kernel matrix $\mathbf{K}$. Inspired by [39], where a stochastic gradient descent algorithm for the GPLVM was proposed, we develop a similar strategy to iteratively approximate the gradient by using a small number of local samples, which supports efficient multi-layer JGPM learning.

Compared with the standard GPLVM algorithm, where all the training samples are taken into account at the same time to compute the gradient, our local learning algorithm involves only a small number of training examples at one time to approximate the gradient locally. First, a reference point $\mathbf{x}_l$ is selected randomly and a neighborhood $\mathbf{X}_L$ centered at $\mathbf{x}_l$ is defined. Then, all the points in the neighborhood $\mathbf{X}_L$ are used to compute the local gradient for updating the latent variable $\mathbf{X}$ locally and the kernel parameters. The local gradient can be represented only by the points

within the neighborhood

$$\frac{\partial \mathcal{L}}{\partial \mathbf{X}_L} = -\left(\mathbf{K}_L^{-1}\mathbf{Y}_L\mathbf{Y}_L^T\mathbf{K}_L^{-1} - D\mathbf{K}_L^{-1}\right) \cdot \frac{\partial \mathbf{K}_L}{\partial \mathbf{X}_L}, \tag{4.9}$$

where $\mathbf{K}_L$ is the kernel matrix for $\mathbf{X}_L$, $\mathbf{Y}_L$ is the corresponding motion data in the neighborhood and D is the dimensionality of motion data. Because the dimensionality of $\mathbf{K}_L$ is small, the computation cost is rather low. Different with [39], there are two special treatments for the local learning in this work. The first one is the integration of our multi-layer topology into the GPLVM-based learning framework, and the other is topology-based neighborhood selection. To incorporate the topology constraints, we use $p(\mathbf{X}_L|\mathbf{W}_L)$ from the LLE energy function in (4.6) to express the local topology constraint, where $\mathbf{W}_L$ is the corresponding weight matrix of latent points within the neighborhood. Every time we randomly choose a latent point as the reference point and repeat the above local gradient operation to optimize one patch of the model with respect to the maximum a posteriori probability (MAP). The posteriori probability is defined as

$$p(\mathbf{X}_L, \boldsymbol{\alpha}, \boldsymbol{\beta}|\mathbf{Y}_L, \mathbf{W}_L) \propto p(\mathbf{Y}_L|\mathbf{X}_L, \boldsymbol{\beta})p(\mathbf{X}_L|\boldsymbol{\alpha})p(\boldsymbol{\alpha})p(\boldsymbol{\beta})p(\mathbf{X}_L|\mathbf{W}_L). \tag{4.10}$$

In each iteration, the computational complexity is $O(M^3)$ for the local learning process, compared with $O(N^3)$ for the original full learning, where M (the number of local neighbors)is far less than N (the number of training data). After sufficient iterations, all the latent points may have been updated many times and the multi-layer JGPM is optimized. Next, we will further discuss our treatment for the neighborhood selection.

In [39], a neighborhood selection strategy of subsampling k neighbors from a larger neighborhood was suggested for allowing sufficient coverage of the latent space. As pointed by the authors, this method may not maintain the neighborhood configuration. In our case, this subsampling method is not suitable as it may interrupt the continuity of latent variables and the layered structure in the multi-layer JGPM.

Thus we have two special considerations for neighborhood selection. First, both within-layer and cross-layer neighbors are involved during the learning process rather than learning each layer separately. Second, because of the given toroidal/cylindrical structure, we can pre-compute a set of neighbors to have sufficient coverage of the latent space, at the same time, to avoid the situation that the gradient estimations are too local to capture the global structure of the latent space. Note that this neighborhood selection for the local learning is different with the neighborhood choosing for the LLE-based topology constraint in Sec. 4.2.4.

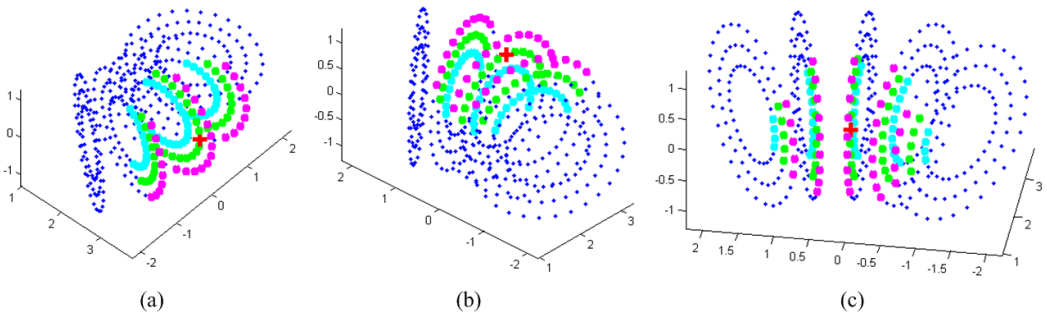


Figure 4.5: Topology-aware neighbor selection for local learning at three locations (a, b, c) in the middle layer: a reference point (in red cross) and its neighbors (in green, magenta and cyan).

To have a trade-off between a sufficient coverage in the latent space and a reasonable computational load, we select no more than 10% of the total training data points according to the Euclidean distance given the multi-layer structures to determine the neighborhood for each reference point. This topology-based neighbor selection will lead to a *topology-aware local learning* process that ensures the learned manifold structure complies with the topological prior. Fig. 4.5 exhibits that for a exemplificative point (in red cross) in the middle layer, neighbors (in green, magenta and cyan) with different pose/gait/scaling indexes are included in its neighborhood. This reveals that both the within-layer and cross-layer constraints are involved during the topology-aware local learning.

4.4 Experimental Results

In this section, we evaluate the proposed multi-layer JGPMs by comparing it with the single-layer JGPM introduced in Chapter 3 and LL-GPDM [58] in terms of three aspects, i.e., latent space illustration, quantitative model validation and qualitative motion synthesis.

4.4.1 Experiment Setting

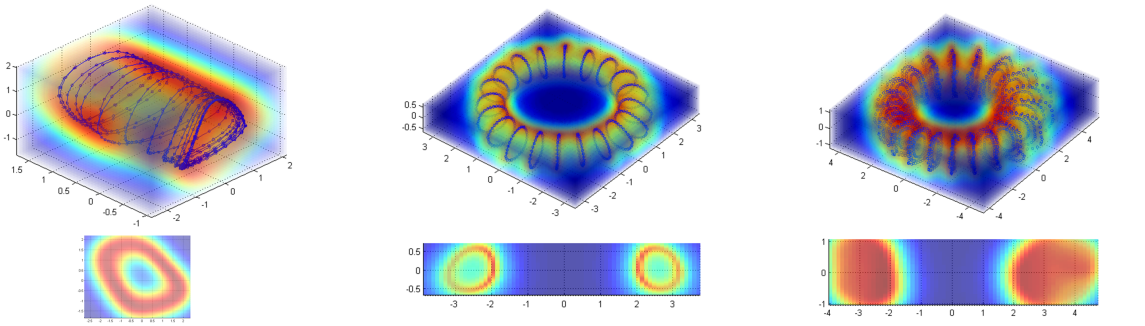
We chose 20 walking sequences from the CMU Mocap Library [34] as the original training data, each of which contains 30 poses downsampled from one walking cycle. Without loss of generality, we consider two scaling factors (0.4 and 1.25) to triple the size of training data according to the *training data diversification* as defined in (4.1). Then we have 60 gaits (scaled from 20) and each gait includes 30 poses, that is there are 1800 data points in the three-layer JGPM.

For the LLE-based topology constraints, considering the computation complexity, we select 16 (10 from within-layer and 6 from cross-layer) neighbors for a reference point as shown in Fig. 4.4. For a point on the middle layer, 3 cross-layer neighbors are selected from each of the outer and inner layers. For a point on the outer layer, 6 cross-layer neighbors are from the middle layer only, while for a point on the inner layer, 6 cross-layer neighbors are from the middle layer only. In *topology-aware local learning*, we select 120 nearest neighbors for each reference point.

4.4.2 Latent Space Illustration

First, we compared the multi-layer JGPM with JGPM and LL-GPDM by illustrating the volumetric representation of their latent space in Fig. 4.6, where the color indicates the prediction confidence (the warmer colors, the higher confidence of motion reconstruction). LL-GPDM has a cylinder-like latent structure, but it only represents the pose manifold explicitly and treats the gait variable implicitly. Both JGPM and

multi-layer JGPM achieved a smooth, compact and physically meaningful latent space that is expected for the human motion modeling. From the cross-section view, it is obvious that multi-layer JGPM has larger high-confidence areas than the other two, implying its more general motion modeling capability. It is expected that multi-layer JGPM is more flexible and robust for motion synthesis and pose estimation. Next, we will evaluate the multi-layer JGPM in terms of motion interpolation, reconstruction and filtering, where both the "toroidal" and "cylindrical" versions are considered to shed some light on the selection of the topology prior for manifold learning.



(a) LL-GPDM and its cross-section view (b) the original JGPM and its cross-section view (c) multi-layer JGPM and its cross-section view

Figure 4.6: Volumetric visualization of prediction confidence in latent spaces; warmer colors, (i.e., red) depict higher confidence of motion reconstruction. (a) LL-GPDM (b) JGPM (c) multi-layer JGPM.

4.4.3 Quantitative Performance

To verify the advantage of the proposed multi-layer JGPM, we quantitatively compare it with the single-layer JGPM and LL-GPDM in terms of three specific tasks, i.e. motion interpolation, motion reconstruction and motion filtering, by employing the same model validation technique used in Chapter 3. The objective of interpolation is to synthesize a new motion sequence from unknown subjects (not from the training subjects), that of reconstruction is to recover the full-body motion from partial-body motion (some joints are missing), and that of filtering is to denoise noisy motion data

from unknown subjects. These experiments help us comprehensively understand the performance of various modeling algorithms.

Motion Interpolation:

We chose twenty walking sequences which are different with the training data from the CMU Mocap Library as our original unknown test data for motion interpolation. Each test sequence has 30 poses downsampled from one walking cycle. Then, as defined in (4.1), we generated four sets of simulated motion data by using four scalars 1.25, 0.667, 0.5 and 0.4, which represent a series of motion ranges. We notice scalars 0.667 and 0.5 are different with those (0.4 and 1.25) used for training data diversification. In addition, we also acquired two sets of real long stride sequences from CMU Mocap dataset (Subject No.7, trail No.11 and Subject No.8, trail No.5). We used a validation method described in [106], by which new motion data were interpolated to represent the unknown test data from a GPLVM-based motion model, and we applied this method to all models. We computed the averaged 3D joint position errors (mm) between the estimated motions and ground truth ones. The interpolation results are illustrated in Fig. 4.7.

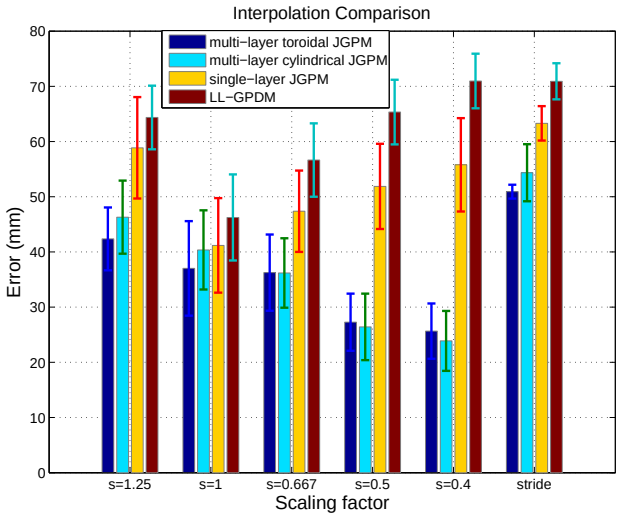


Figure 4.7: Comparison of interpolation results.

It is shown that the multi-layer JGPMs (toroidal and cylindrical) are more accurate than the original JGPM and LL-GPDM to represent the unknown data, especially when the motion data with larger or smaller scaling factors, which implies the superior representative capability and flexibility of multi-layer JGPM. Fig. 4.8 visualizes the motion interpolation results of some simulated test data using stick man, where the red points represent the ground-truth and the blue points are the interpolation results. Also, Fig. 4.9 shows the interpolation results of real stride motion sequence. Obviously, the multi-layer JGPM has better performance.

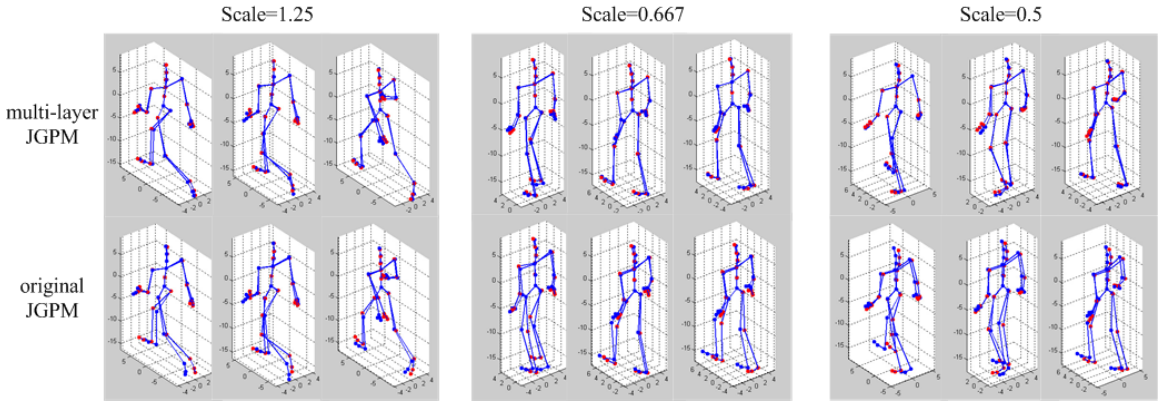


Figure 4.8: Motion interpolation results, where the red and blue points represent the ground-truth and estimated results respectively.

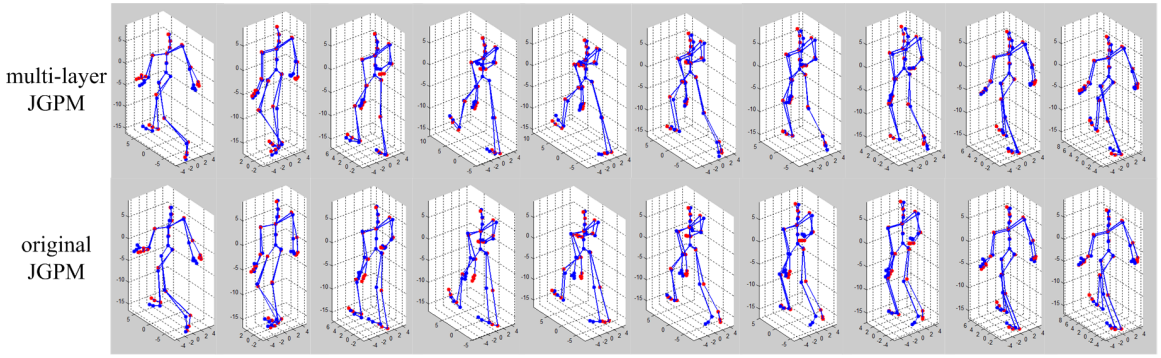


Figure 4.9: Motion interpolation results of the real stride sequences, where the red and blue points represent the ground-truth and estimated results respectively.

Motion Reconstruction:

In this paper, we term motion reconstruction as a missing data recovering problem. We studied three reconstruction cases, i.e., missing the left arm (3 joints), missing the left leg (3 joints) and missing the left-side body(6 joints). We utilized the same test data with motion interpolation and similar model validation algorithm to recover the full-body motions (17 joints) from four different learned models respectively. We then computed the averaged 3D joint position errors (mm) between the reconstructed motions and ground truth ones. The reconstruction results are depicted in Fig. 4.10. It is still demonstrated that the multi-layer JGPM (toroidal and cylindrical) provides better performance than the original JGPM and LL-GPDM to recover the full-body motion from partial data, especially when the motion data with larger or smaller scaling factors.

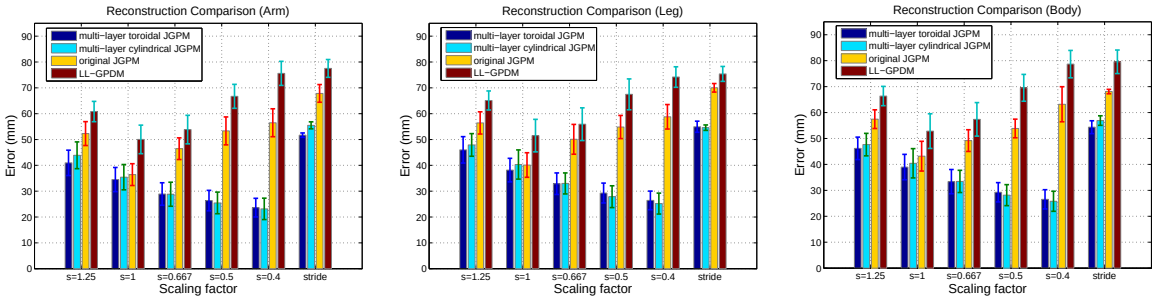


Figure 4.10: Missing body part interpolation results using multi-layer JGPM (toroidal and cylindrical), the original JGPM and LL-GPDM.

Motion Filtering A better motion model should provide better noise filtering results. In this experiment, we utilized the same unknown test data as we used in the previous experiments to compare the filtering performance of all motion models. For each scaled dataset and stride motion, three noisy motion datasets were generated by adding additive white Gaussian noise (AWGN) at three levels (5%, 10% and 15%) with respect to the standard deviation of each joint angle. The filtering process was repeated by five times using five sets of random noise and then we obtained the mean

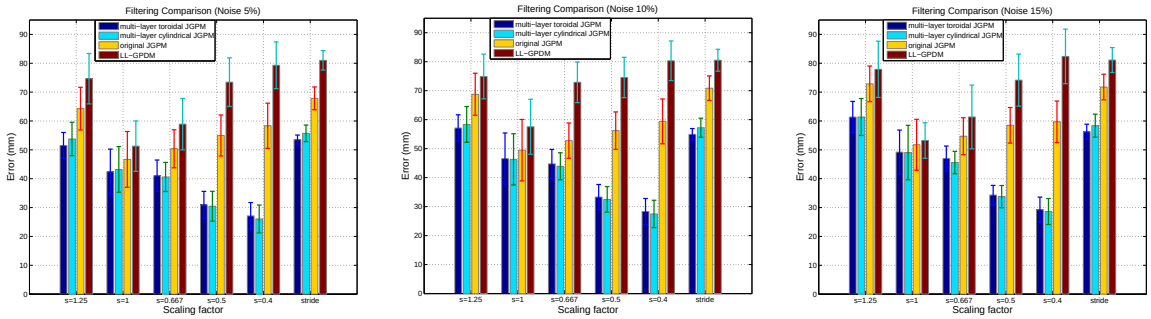


Figure 4.11: Noisy subjects filtering results using multi-layer JGPM (toroidal and cylindrical), the original JGPM and LL-GPDM.

errors for each noise level. Fig.4.11 shows that the multi-layer JGPM (toroidal and cylindrical) is more accurate (less errors) and robust (less standard deviations) than JGPM to filter the unknown motion data in all three noise level as well as all the motion ranges. It is interesting to find that not only for the scaled and real stride motion sequences, but also for the original unscaled motion data ($s = 1$), the proposed multi-layer JGPM demonstrate significant advantages.

4.4.4 Motion Synthesis via Latent Space Sampling

To further evaluate the multi-layer JGPM and original JGPM models, we can sample their latent spaces along certain trajectory and visualize the reconstructed motion data accordingly. In this experiment, we used three sampling trajectories, i.e., a horizonal straight line, a large circular spiral outside and a small circular spiral inside, as shown in Fig. 4.12. For the first trajectory, we expect there should be a gradual motion range increase under the same pose. For the latter two trajectories, we expect to see two walking sequences with two extreme motion ranges. As shown in Fig. 4.12, the original JGPM offer limited capability to synthesize humanoid walking motion with different styles, especially very large or small motion ranges. The distortion becomes more severe when samples are away from the learned manifold structure. Compared with JGPM, the multi-layer JGPM has great flexibility to synthesize hu-

manoid walking motion with various styles.

4.5 Discussion

In this paper, we have proposed multi-layer joint gait-pose manifolds (JGPM) in order to enhance the representability and the flexibility of the human motion model. We mainly focus on human gait motion that embraces a variety of walking styles. There are also some limitations of our proposed algorithm, which will guide our future research. First, our proposed model is limited to motions that share some similar pattern that is important to learn a smooth manifold. Second, we assume the full body motion dyanmics could be scaled by only one scalar, which may lead to some systematic error. It is possible different scalars are needed at different body segments. However, without the exact anthropometric measuring and biomechanics evidence, it is challenging to find the segment-specific scaling factors. These two limitations could be mitigated by constructing a hierarchical model [114], where our proposed multi-layer JGPM is learned on many different motion types/styles and connected through a multi-level latent model or by building a part-level model [115] to represent the motion of different human segments. Also, to further speed up the learning procedure, some other sparsification methods [110, 111, 112, 113] designed for lager datasets or incremental learning [116, 117] which is suitable for fast sequentially online learning could be integrated with our topology-aware local learning algorithm. Although our modeling method is designed for a specific type of motion, i.e., gait, the multi-layer latent structure as well as the two key techniques are general and could potentially apply to other human motion types or other dataset, like face expression and handwriting.

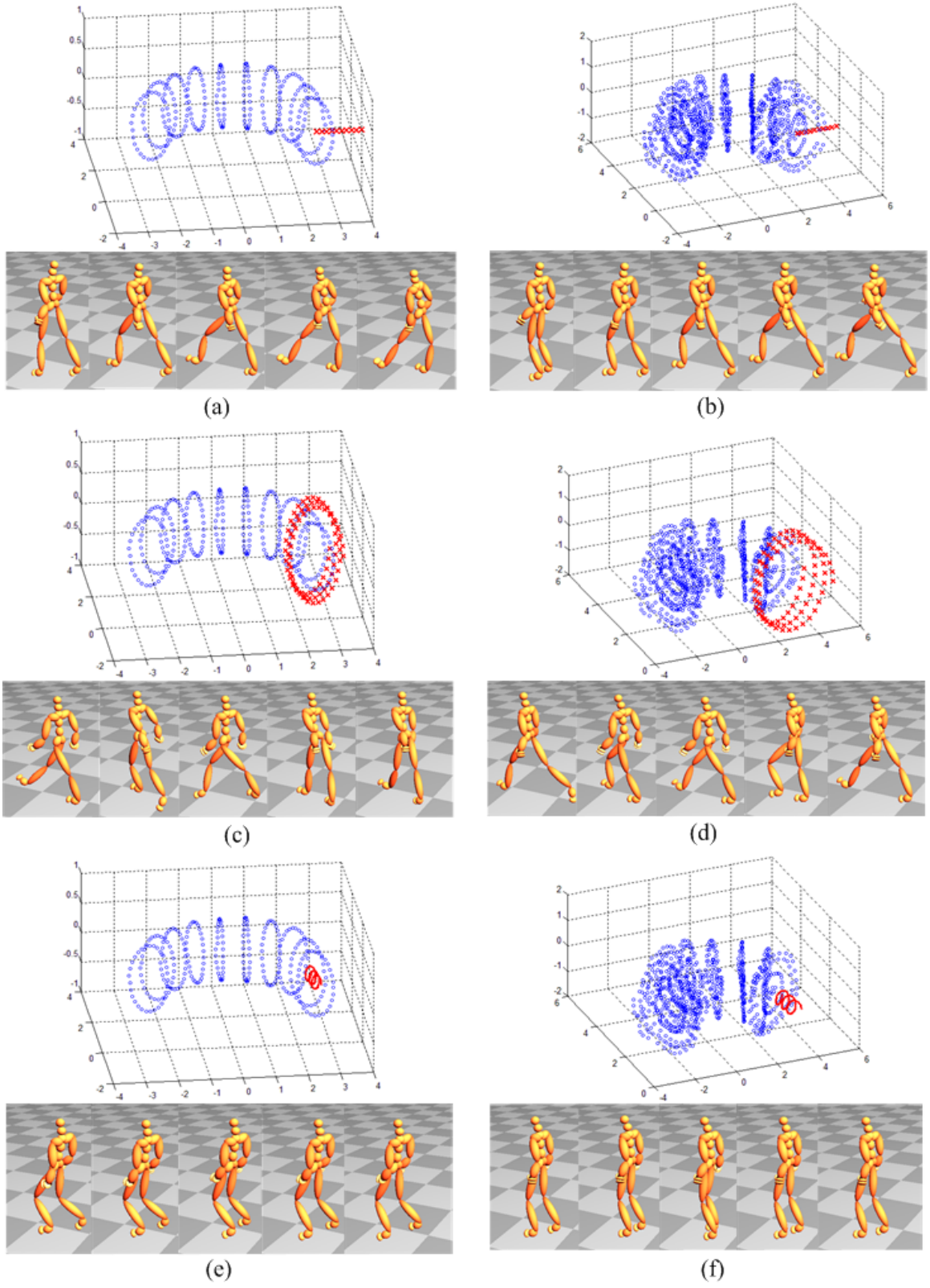


Figure 4.12: Motion synthesis by sampling JGPM (left) and the multi-layer JGPM (right).

CHAPTER 5

GAUSSIAN KERNEL CORRELATION (GKC)

5.1 Introduction

Registration which aims to transform different data sets into one coordinate system is one of the fundamental research topics in the field of computer vision and image processing. There are many applications of registration, e.g., medical imaging, brain mapping, image stitching, 3D reconstruction, augmented reality, etc. In this research, we mainly focus on the articulated pose estimation, especially for the human body and hand pose estimation using a registration method, i.e., generalized Gaussian kernel correlation (GKC).

According to how the template and the target are matched, registration approaches can be classified into two major categories, i.e., correspondence-based and correspondence-free. The algorithms in the first category iteratively estimate the correspondences and the underlying transformation, such as the Iterative Closest Point (ICP) [65] and the Maximum Likelihood-based density estimation [79, 80, 81, 82]. The algorithms in the second group directly optimize an energy function without involving correspondences, including density alignment [83] and kernel correlation [84]. Different with the density alignment whose energy function is a discrepancy measure using L2 distance, kernel correlation was proposed as a similarity measure in[85, 84] and was used for point set registration. In KC-based registration, both the template and the observation are modeled by kernels and their registration is achieved by maximizing their similarity. The kernel correlation was also applied to the stereo vision-based modeling in [86].

When the kernel function is a Gaussian, there are two unique benefits for registration, i.e., robustness and efficient optimization. First, since the statistical benefits of Gaussian, Gaussian KC (GKC) is as robust as the M-estimator [87], as mentioned in [86]. Second, different from the Maximum Likelihood-based registration using Expectation-Maximization (EM) [80, 81, 82], the closed-form expression of GKC supports a direct gradient-based optimization which is more efficient and robust. However, existing GKC mainly considers the case of univariate (isotropic) Gaussian only with two exceptions (to the best of our knowledge). First, Sum of univariate Gaussians (SoG) was extended to sum of anisotropic Gaussians (SAG) in [75] where the similarity function was evaluated in the projected 2D image space. Our previous work [30] studied anisotropic Gaussians in 3D space and derived a similarity measure between the template and target, represented by multivariate and univariate Gaussians, respectively. In this work, we generalize both approaches by developing a n-dimensional Gaussian KC function which supports a unified similarity measure between two collections of arbitrary univariate / multivariate Gaussian kernels.

5.2 Univariate Gaussian Kernel Correlation

Given two Gaussians centered at points $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2 \in \mathbb{R}^n$, their kernel correlation is defined as the integral of the product of two Gaussian kernels over the n dimensional space [84],

$$KC(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2) = \int_{\mathbb{R}^n} G(\mathbf{x}, \boldsymbol{\mu}_1) \cdot G'(\mathbf{x}, \boldsymbol{\mu}_2) d\mathbf{x}, \tag{5.1}$$

where $\mathbf{x} \in \mathbb{R}^n$, and $G(\mathbf{x}, \boldsymbol{\mu}_1), G'(\mathbf{x}, \boldsymbol{\mu}_2)$ represent the Gaussian kernels centered at the data point $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2$, respectively. Different from [84], where the Gaussian kernel has a standard univariate Gaussian distribution form, we employ an non-normalized Gaussian kernel defined in [118],

$$G^{(u)}(\mathbf{x}, \boldsymbol{\mu}) = \exp(-\frac{||\mathbf{x} - \boldsymbol{\mu}||^2}{2\sigma^2}), \tag{5.2}$$

where (u) represents "univariate" and σ^2 is the variance. The non-normalized Gaussian kernel can lead to a more controllable and meaningful kernel correlation between two Gaussians with large differences in variance, because the non-normalized G and G' have a similar scale even if their variances σ_1, σ_2 are largely distinct, as shown in Fig. 5.1. This features allows us to use large variance in the template to reduce the number of Gaussian kernels, even if the variances of Gaussian kernels are very small in the observed point cloud.

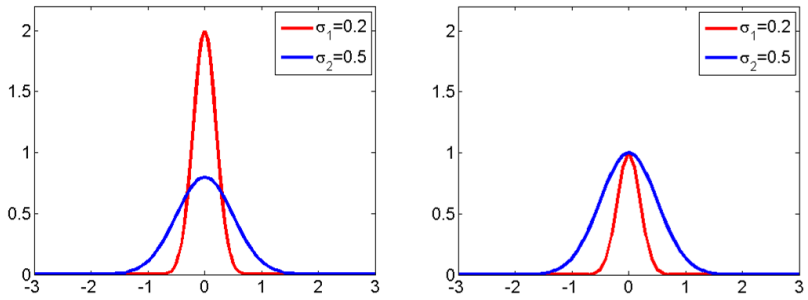


Figure 5.1: The comparison of normalized (left) and non-normalized (right) Gaussian kernels with the same variances σ_1, σ_2.

Plugging (5.2) in (5.1), it is straightforward to have the kernel correlation of two (non-normalized) univariate Gaussians at $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$,

$$UKC(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2) = \left(2\pi \frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 + \sigma_2^2}\right)^{\frac{n}{2}} \exp\left(-\frac{||\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2||^2}{2(\sigma_1^2 + \sigma_2^2)}\right). \tag{5.3}$$

This equation is to measure the similarity between two univariate Gaussians. When two Gaussians are close to each other and have similar variances, their similarity becomes larger; otherwise, it becomes smaller.

5.3 Multivariate Gaussian Kernel Correlation

In this section, we generalize the original Gaussian kernel correlation in [84] from two aspects. First, we extend the univariate Gaussian to the multivariate one and derive a unified GKC function between two Gaussians in n dimensional space. Second,

we provide a more general kernel correlation between two collections of Gaussian kernels, both of which can be composed by univariate/multivariate Gaussian kernels (Fig. 5.3 (a-c)) or even the mixed model (Fig. 5.3 (d)).

If the variance σ^2 is extended to the covariance matrix Σ, we have the non-normalized multivariate Gaussian kernel form,

$$G^{(m)}(\mathbf{x}, \boldsymbol{\mu}) = \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})\right). \tag{5.4}$$

Obviously, when Σ is a diagonal matrix and the diagonal entries are identical, the equation (5.4) will degenerate to (5.2). We illustrate the geometrical expression of multivariate Gaussian in 3D space, as shown in Fig. 5.2.

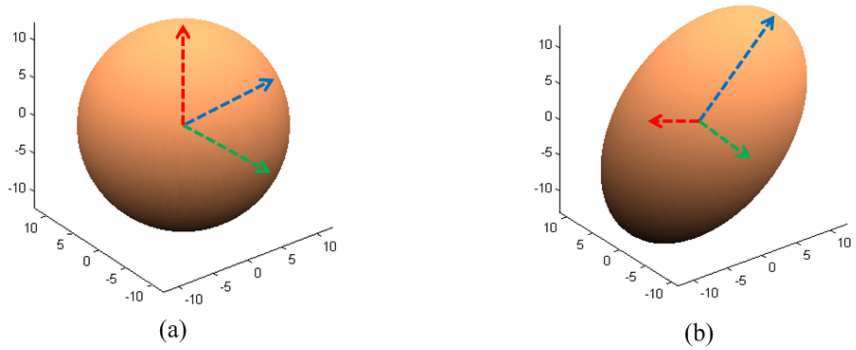


Figure 5.2: The geometrical expression of univariate and multivariate Gaussian in 3D space.

Now, we re-write (5.1) using (5.4) to derive the generalized Gaussian kernel correlation, which is not as straightforward as (5.3). The proof of the unified Gaussian kernel correlation in (5.5) is listed below. Given two non-normalized Gaussian kernels centered at two points $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2$,

$$G_1^{(m)}(\mathbf{x}, \boldsymbol{\mu}_1) = \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_1)^T\Sigma_1^{-1}(\mathbf{x}-\boldsymbol{\mu}_1)\right)$$
$$G_2^{(m)}(\mathbf{x}, \boldsymbol{\mu}_2) = \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_2)^T\Sigma_2^{-1}(\mathbf{x}-\boldsymbol{\mu}_2)\right),$$

we aim to derive their kernel correlation $KC_m(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2)$ which is represented as,

$$MKC(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2) = \int_{\mathbb{R}^n} G_1^{(m)}(\mathbf{x}, \boldsymbol{\mu}_1) \cdot G_2^{(m)}(\mathbf{x}, \boldsymbol{\mu}_2) d\mathbf{x}.$$

We re-write $G_1^{(m)}(\mathbf{x}, \boldsymbol{\mu}_1)$ and $G_2^{(m)}(\mathbf{x}, \boldsymbol{\mu}_2)$ in canonical notation as,

$$G_1^{(m)}(\mathbf{x}, \boldsymbol{\mu}_1) = \exp\left(-\frac{1}{2}\mathbf{x}^T\Sigma_1^{-1}\mathbf{x} + (\Sigma_1^{-1}\boldsymbol{\mu}_1)^T\mathbf{x} - \frac{1}{2}\boldsymbol{\mu}_1^T\Sigma_1^{-1}\boldsymbol{\mu}_1\right)$$
$$G_2^{(m)}(\mathbf{x}, \boldsymbol{\mu}_2) = \exp\left(-\frac{1}{2}\mathbf{x}^T\Sigma_2^{-1}\mathbf{x} + (\Sigma_2^{-1}\boldsymbol{\mu}_2)^T\mathbf{x} - \frac{1}{2}\boldsymbol{\mu}_2^T\Sigma_2^{-1}\boldsymbol{\mu}_2\right)$$

Therefore,

$$\begin{aligned}
G_1^{(m)} \cdot G_2^{(m)} &= \exp\Big(-\frac{1}{2}\mathbf{x}^T(\Sigma_1^{-1} + \Sigma_2^{-1})\mathbf{x} + (\Sigma_1^{-1}\boldsymbol{\mu}_1 + \Sigma_2^{-1}\boldsymbol{\mu}_2)^T\mathbf{x} \\
&\quad -\frac{1}{2}\boldsymbol{\mu}_1^T\Sigma_1^{-1}\boldsymbol{\mu}_1 - \frac{1}{2}\boldsymbol{\mu}_2^T\Sigma_2^{-1}\boldsymbol{\mu}_2\Big) \\
&= \exp\Big(-\frac{1}{2}\mathbf{x}^T(\Sigma_1^{-1} + \Sigma_2^{-1})\mathbf{x} + \big((\Sigma_1^{-1} + \Sigma_2^{-1})\boldsymbol{\mu}^*\big)^T\mathbf{x} \\
&\quad -\frac{1}{2}\boldsymbol{\mu}^{*T}(\Sigma_1^{-1} + \Sigma_2^{-1})\boldsymbol{\mu}^* + \frac{1}{2}\boldsymbol{\mu}^{*T}(\Sigma_1^{-1} + \Sigma_2^{-1})\boldsymbol{\mu}^* \\
&\quad -\frac{1}{2}\boldsymbol{\mu}_1^T\Sigma_1^{-1}\boldsymbol{\mu}_1 - \frac{1}{2}\boldsymbol{\mu}_2^T\Sigma_2^{-1}\boldsymbol{\mu}_2\Big) \\
&= \exp\Big(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}^*)^T(\Sigma_1^{-1} + \Sigma_2^{-1})(\mathbf{x} - \boldsymbol{\mu}^*)\Big) \\
&\quad \cdot \exp\Big(-\frac{1}{2}\big(\boldsymbol{\mu}_1^T\Sigma_1^{-1}\boldsymbol{\mu_1} - \boldsymbol{\mu}^{*T}(\Sigma_1^{-1} + \Sigma_2^{-1})\boldsymbol{\mu}^* \\
&\quad +\boldsymbol{\mu}_2^T\Sigma_2^{-1}\boldsymbol{\mu_2}\big)\Big),
\end{aligned}$$

where

$$\begin{aligned}
\boldsymbol{\mu}^* &= (\Sigma_1^{-1} + \Sigma_2^{-1})^{-1}(\Sigma_1^{-1}\boldsymbol{\mu}_1 + \Sigma_2^{-1}\boldsymbol{\mu}_2) \\
&= \Sigma_1(\Sigma_1 + \Sigma_2)^{-1}\boldsymbol{\mu}_2 + \Sigma_2(\Sigma_1 + \Sigma_2)^{-1}\boldsymbol{\mu}_1.
\end{aligned}$$

Then, we have

$$\begin{aligned}
G_1^{(m)} \cdot G_2^{(m)} &= \exp\Big(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}^*)^T(\Sigma_1^{-1} + \Sigma_2^{-1})(\mathbf{x} - \boldsymbol{\mu}^*)\Big) \\
&\quad \cdot \exp\left(-\frac{1}{2}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T(\Sigma_1 + \Sigma_2)^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)\right),
\end{aligned}$$

According to the Gaussian integral

$$\int_{\mathbb{R}^n} \exp(-\frac{1}{2}\mathbf{x}^T\Sigma\mathbf{x})d\mathbf{x} = \sqrt{\frac{(2\pi)^n}{|\Sigma|}},$$

we have,

$$\begin{aligned} MKC(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2) &= \int_{\mathbb{R}^n} \exp\Big(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}^*)^T(\Sigma_1^{-1}+\Sigma_2^{-1})(\mathbf{x}-\boldsymbol{\mu}^*)\Big) \\ &\quad \cdot \exp\left(-\frac{1}{2}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)^T(\Sigma_1+\Sigma_2)^{-1}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)\right) d\mathbf{x} \\ &= \sqrt{\frac{(2\pi)^n}{|\Sigma_1^{-1}+\Sigma_2^{-1}|}} \cdot \\ &\quad \exp\left(-\frac{1}{2}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)^T(\Sigma_1+\Sigma_2)^{-1}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)\right). \end{aligned}$$

Finally, we have the kernel correlation of two n dimensional multivariate Gaussian kernels which are centered at points $\boldsymbol{\mu_1}, \boldsymbol{\mu_2}$ and modeled by the covariance matrices Σ_1, Σ_2 respectively,

$$\begin{aligned} MKC(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2) &= \sqrt{\frac{(2\pi)^n}{|\Sigma_1^{-1}+\Sigma_2^{-1}|}} \cdot \\ &\quad \exp\left(-\frac{1}{2}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)^T(\Sigma_1+\Sigma_2)^{-1}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)\right). \end{aligned} \tag{5.5}$$

Different from statistical correlation to represent the proximity of two distributions in the statistics, our kernel correlation, where the non-normalized Gaussian kernels are involved, is defined as a kind of energy to measure the similarity of two parametrical models. In other words, the energy becomes larger as the two kernel models become closer and more similar to each other.

5.4 Generalized GKC for Two Collections of Gaussian Kernels

Several Gaussian kernels which are centered at a set of points $\Omega = \{\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_m\}$ can be combined as a sum of Gaussian kernels $\mathcal{K}$,

$$\mathcal{K} = \sum_{i=1}^{m} G(\mathbf{x}, \boldsymbol{\mu}_i). \tag{5.6}$$

Given two collections of Gaussian kernels $\mathcal{K}_A$ and $\mathcal{K}_B$ composed by M and N Gaussian kernels respectively, their kernel correlation is defined as,

$$\begin{aligned} MKC(\mathcal{K}_A, \mathcal{K}_B) &= \int_{\mathbb{R}^n} \sum_{i=1}^{M} \sum_{j=1}^{N} G(\mathbf{x}, \boldsymbol{\mu}_i^{(A)}) G'(\mathbf{x}, \boldsymbol{\mu}_j^{(B)}) d\mathbf{x} \\ &= \sum_{i=1}^{M} \sum_{j=1}^{N} MKC(\boldsymbol{\mu}_i^{(A)}, \boldsymbol{\mu}_j^{(B)}), \end{aligned} \tag{5.7}$$

where $MKC(\boldsymbol{\mu}_i^{(A)}, \boldsymbol{\mu}_j^{(B)})$ has been derived in (5.5). It worth noting that $\mathcal{K}_A$ and $\mathcal{K}_B$ can be composed by univariate Gaussians (Fig. 5.3 (a)), multivariate ones (Fig. 5.3 (b,c)) or mixed ones (Fig. 5.3 (d)). Consequently, we obtain a unified kernel correlation function in (5.7) to evaluate the similarity between any pairwise combination of univariate and multivariate SoG models, as shown in Fig. 5.3. When the covariance matrices in $\mathcal{K}_B$ degenerate to variances in the 3D space, the degenerated equation (5.7) will be equivalent to the SoG↔GSoG similarity in [30]. Further, if the covariance matrices in $\mathcal{K}_A$ degrade to variances in 3D, the equation (5.7) will become the SoG↔SoG similarity in [29, 73, 74]. Both degenerations imply that our kernel correlation functions in (5.5) and (5.7) generalize all the previous SoG-based methods.

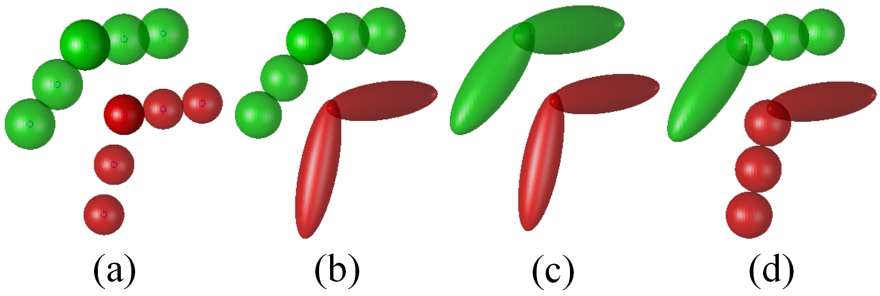


Figure 5.3: The illustration of the sum of Gaussian kernels $\mathcal{K}_A$ (red) and $\mathcal{K}_B$ (green) in 3D with four cases: (a) SoG-SoG, (b) SoG-GSoG, (c) GSoG-GSoG, (d) mixed model-mixed model.

5.5 Discussion

The derived equation 5.5 is partially coincident with the formulation of the multivariate mean integrated squared error (MISE) in [119]. However, there is no explicit formulation derivation provided. Since we use non-normalized Gaussian kernels, the main difference between our generalized GKC and their MISE is the coefficient before the exponential function. Their MISE of two distributions has statistical meaning, but our GKC emphasizes a concept of energy to measure the similarity. This generalized GKC provides us a fundamental tool for us to construct our subject-specific shape modeling and articulated pose estimation algorithms, which will be presented in the next two chapters.

CHAPTER 6

ARTICULATED GKC FOR SHAPE MODELING

6.1 Introduction

A good shape model not only captures shape variability accurately, but also facilitates the data matching efficiently for pose tracking. In this chapter, we first embed an articulated skeleton into a collection of Gaussian kernels where quaternion-based 3D rotations are involved to represent the transformation between two segments along the skeleton. Then, based on the generalized GKC in equation (5.7), a segment-scaled articulated Gaussian kernel correlation (AGKC) is proposed to balance the effect of each segment in the articulated structure. Using the segment-scaled AGKC as an energy function, we propose an effective and efficient subject-specific shape modeling method, where a LLE-based topology constraint is developed as a regularization term.

6.2 Articulated Shape Modeling with Gaussian Kernels

In this work, we use the full-body human and hands as examples to present the Gaussian kernels-based articulated shape model, as shown in Fig. 6.1. For the task of human pose estimation, the body template comprises a kinematic skeleton (Fig. 6.1 (a)) and a Gaussian kernel-based shape model $\mathcal{K}_A$. Fig. 6.1 (c) and (d) exhibit the univariate and multivariate Gaussians represented body shape models and their volumetric density comparison in the projected 2D image. The shape models for hand and their volumetric density comparison are shown in Fig. 6.1 (e) and (f). We can observe that the density map of multivariate Gaussians has a more distinct and

smooth silhouette than that of univariate Gaussians, revealing two major benefits of using multivariate ones to approximate an articulated object. First, the smooth and continuous density of multivariate Gaussians facilitates the optimizer to achieve more accurate pose estimation results. Second, the anatomical landmarks (i.e. body/finger joints) have clear definitions in the multivariate case. Our previous study in [30] has also shown the better flexibility and adaptability of multivariate Gaussians for shape modeling.

In the following, our discussion is mainly focused on the human model which is also applicable to hands and other articulated objects. We denote $\widetilde{\mathcal{K}_A}$ as a standard T-pose template as shown in Fig. 6.1 (d). The kinematic skeleton is constructed by a tree-structured chain, as illustrated in Fig. 6.2.

Each rigid body segment has its local coordinate system that can be transformed to the world coordinate system via a 4×4 transformation matrix T_l,

$$T_l = T_{par(l)} R_l, \tag{6.1}$$

where R_l denotes the local transformation from body segment S_l to its parent $par(S_l)$. Since each segment is attached on its corresponding body joint marked as red stars in Fig. 6.1 (a), the index l is used in both the body joint and its associated segment. In this work, each joint in the body has 3 degrees of freedom (DoF) rotation, and the joints marked with the red circles and stars in the hand model (Fig. 6.1 (b)) have 1 DoF and 3 DoF rotation, respectively. If l is the root joint (the hip joint), T_{root} is the global transformation of the whole body. Given a transformation matrix T_l, the center of k_{th} Gaussian kernel in the segment S_l at the T-pose $\widetilde{\boldsymbol{\mu}}_{l,k}$ can be transferred to its corresponding position in the world coordination,

$$\boldsymbol{\mu}_{l,k} = T_l \widetilde{\boldsymbol{\mu}}_{l,k}. \tag{6.2}$$

Accordingly, the local transformation R at each body joint and T_{root} define a specific pose. Since the translation between two segments is pre-defined, only the rotation

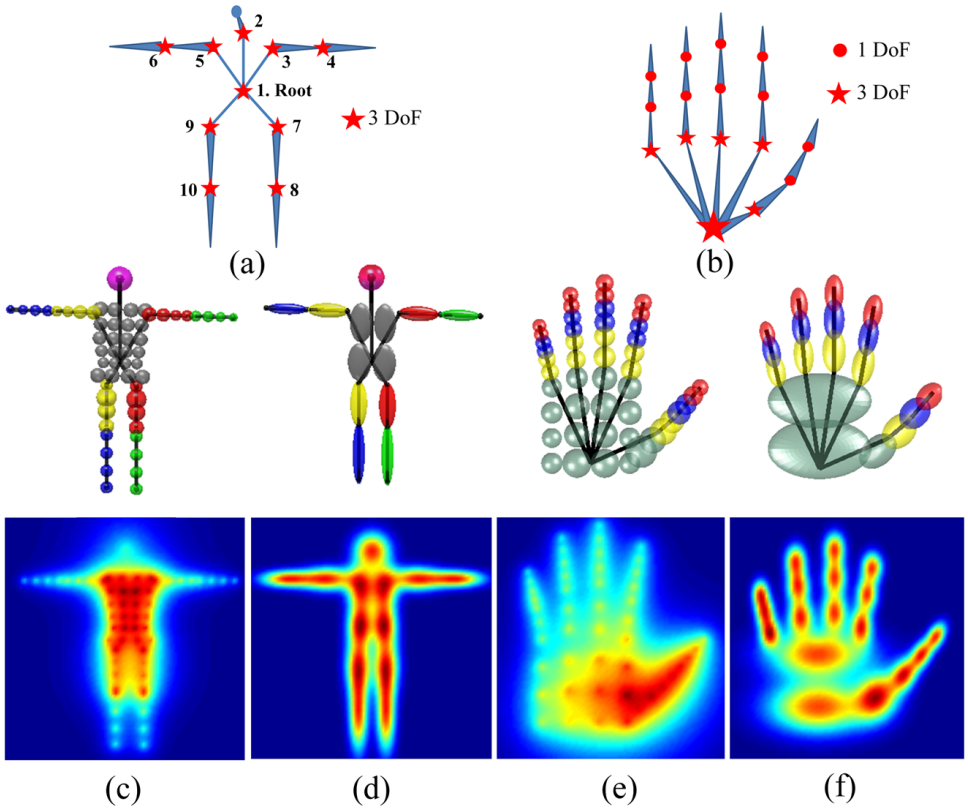


Figure 6.1: (a) and (b) show the skeletons of human and hand respectively. (c) and (d) illustrate the univariate and multivariate Gaussians represented body models and their volumetric density comparison in the projected 2D image. (e) and (f) are the hand shape model and their volumetric density in 2D. To obtain the density map, the variance of each univariate Gaussian has been manually optimized to depict a decent color map. Obviously, the silhouette of multivariate Gaussians is more distinct, compact and smooth than that of univariate ones.

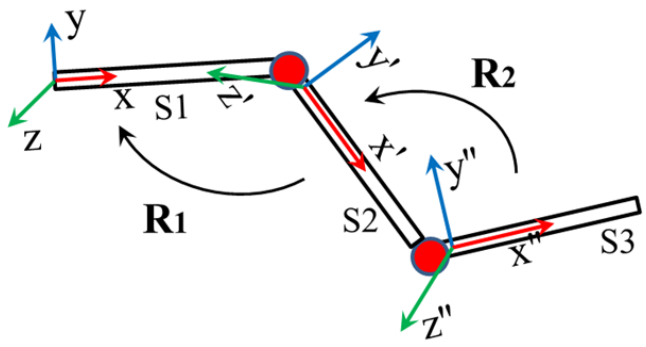


Figure 6.2: The illustration of a kinematical chain structure and the coordination transformation from the child segment to its parent segment, i.e., $S_3 \rightarrow S_2$ via R_2 and $S_2 \rightarrow S_1$ via R_1.

is to be estimated in each R. In this work, we express a 3D joint rotation as a normalized quaternion due to its continuity which can facilitate the gradient-based optimization. Here, we have L joints ($L = 10$, marked as red stars in Fig. 6.1 (a)), each of which allows a 3 DoF rotation represented by a quaternion vector of four elements. Also, there is a global translation at the hip (root) joint. As a result, we totally have 43 parameters/dimensions in a full-body pose represented by $\mathbf{\Theta}$. In the hand model, since 1 DoF rotation is controlled by two elements of a quaternion, there are totally 47 pose parameters. Similar to (6.2), given the body model at T-pose $\widetilde{\mathcal{K}_A}$, the deformed model under pose $\mathbf{\Theta}$ is,

$$\begin{aligned} \mathcal{K}_A &= \widetilde{\mathcal{K}_A}(\mathbf{\Theta}) \\ &= \sum_{i=1}^{M} G(\mathbf{x}, \widetilde{\boldsymbol{\mu}}_i^{(A)}(\mathbf{\Theta})). \end{aligned} \tag{6.3}$$

Consequently, the Gaussian kernels are embedded into an articulated skeleton and controlled by the quaternion-based pose variable $\mathbf{\Theta}$. This articulated Gaussian kernel-based shape representation is general and can be applied to any other articulated shape models. Re-writing (5.7) using (6.3), we explicitly obtain the articulated Gaus-

sian kernel correlation as,

$$MKC(\widetilde{\mathcal{K}_A}(\mathbf{\Theta}), \mathcal{K}_B) = \sum_{i=1}^{M}\sum_{j=1}^{N} MKC(\widetilde{\boldsymbol{\mu}}_i^{(A)}(\mathbf{\Theta}), \boldsymbol{\mu}_j^{(B)}), \tag{6.4}$$

where $MKC(\widetilde{\boldsymbol{\mu}}_i^{(A)}(\mathbf{\Theta}), \boldsymbol{\mu}_j^{(B)})$ can be calculated in (5.5). As a similarity measure, the analytical representation of our articulated kernel correlation in (6.4) become the main part of our objective function. As a result, the problem of articulated pose estimation is converted to finding the optimal $\mathbf{\Theta}$ by which the deformed template $\widetilde{\mathcal{K}_A}(\mathbf{\Theta})$ has the maximum kernel correlation with $\mathcal{K}_B$, i.e., Gaussian Kernel-based representation of an observed point cloud. Next, we further propose a new segment-scaled Gaussian kernel correlation to balance the effect of each segment in an articulated structure.

6.3 Segment-scaled Gaussian Kernel Correlation

The Gaussian kernel correlation $MKC(\widetilde{\mathcal{K}_A}(\mathbf{\Theta}), \mathcal{K}_B)$ can be evaluated according to (6.4) and (5.5). In practice, we found that the kernel correlation from larger segments (e.g. torso in the human body or palm in the hand) could dominate the energy function, overshadowing contributions from small segments. This bias may trap the optimizer in a wrong local minimum, since the gradient direction is also mostly affected by the large segments. To balance the energy contributions from different segments, we further upgrade (6.4) to balance the influence of each articulated segment, referred as "segment-scaled Kernel Correlation". Specifically, the kernel correlation from body segment S_l is weighted by a coefficient $\frac{1}{\omega_l}$ as,

$$sMKC(\widetilde{\mathcal{K}_A}(\mathbf{\Theta}), \mathcal{K}_B) = \sum_{l=1}^{L}\frac{1}{\omega_l}\sum_{k=1}^{K_l}\sum_{j=1}^{N} MKC(\widetilde{\boldsymbol{\mu}}_{l,k}^{(A)}(\mathbf{\Theta}), \boldsymbol{\mu}_j^{(B)}), \tag{6.5}$$

where K_l is the number of Gaussian kernels in the segment S_l (totally we have L segments with the equality $K_1 + \cdots + K_l + \cdots + K_L = M$), and $\frac{1}{\omega_l}$ means the weight of the corresponding segment S_l. Without loss of generality, we calculate ω_l as the

integral of all the Gaussian kernels in the segment S_l,

$$\begin{aligned}\omega_l &= \int_{\mathbb{R}^n} \sum_{k=1}^{K_l} G(\mathbf{x}, \widetilde{\boldsymbol{\mu}}_k) d\mathbf{x} \\ &= \sum_{k=1}^{K_l} \sqrt{\frac{(2\pi)^n}{|\Sigma_k^{-1}|}},\end{aligned} \tag{6.6}$$

where ω_l denotes the volumetric measure of the segment S_l. In other words, the larger body segment, the greater value of ω_l, and the smaller weight it has. In this way, we balance the contribution of each body segment for the holistic kernel correlation using a given subject-specific body shape. Meanwhile, the value of ω_l can be calculated off-line without affecting the online performance.

6.4 Subject-specific Shape Model Learning

We propose an efficient two-step approach to estimate the subject-specific shape model that is represented by a multivariate SoG along with a certain-sized skeleton. We first use an auxiliary univariate SoG model (order 57) for skeleton/shape learning, and then we convert it to the final shape model of a lower order multivariate SoG (order 13) which will be used for pose estimation and tracking. This approach effectively reduces the space of SoG parameters and still takes advantage of the multivariate SoG for shape modeling.

In this first step, we choose one template pose which has a clear articulated structure to support accurate estimation of bone length and body shape for each new subject, as shown in Fig. 6.3. We want to loose the rigid body constraints and to allow free movement of each Gaussian kernel for better adapting to the observation under a "neutral" pose when the subject's four limbs are fully stretched. A set of SoG parameters (totally $57 \times 4 = 228$), $\mathbf{\Pi}$, which defines the location and variance of each univariate Gaussian is optimized by maximizing the KC function defined (6.5). However, some Gaussian kernels from different body parts could be blended near joints, as shown in Fig. 6.3 (b). To avoid this problem, we augment a Local Linear

Embedding (LLE)-based topology constraint [22] which aims to preserve the articulated structure in the SoG-based shape representation. The new objective function for the subject-specific shape modeling is defined as:

$$\hat{\mathbf{\Pi}} = \arg\min_{\mathbf{\Pi}} \Big\{ -UKC(\widetilde{\mathcal{K}_A}(\mathbf{\Pi}), \mathcal{K}_B) + \lambda \sum_{i=1}^{M} ||\boldsymbol{\mu}_i - \sum_{j \in \tau_i} w_{ij} \boldsymbol{\mu}_j||^2 \Big\}, \tag{6.7}$$

where $\boldsymbol{\mu}_i$ is the mean of the ith Gaussian in the body model; τ_i represents the K nearest neighbors ($K = 4$ in this work) of the ith Gaussian; w_{ij} is the LLE weight; λ controls the weight of the LLE term. This objective function can be optimized by an nonlinear optimizer, like Quasi-Newton [120]. The details of the gradient of the energy function in each dimensionality can be referred to the derived expression in the Chapter 7.4. The subject-specific SoG-based body model is shown in Fig. 6.3 (c), where it is straightforward to calculate the limb lengths.

In the second step, we map each Gaussian kernel in the univariate SoG to a corresponding body segment and then compute the covariance matrix of each Gaussian kernel in the multivariate SoG using a pre-defined relationship. For example, six Gaussian kernels on the top-left part of the torso can be mapped to one anisotropic Gaussian in the corresponding position in the multivariate SoG-based model. Given this mapping, we use the mean of the six univariate Gaussians as the mean of the corresponding multivariate Gaussian and we employ PCA to estimate the covariance matrix using the first three eigenvectors and associated eigenvalues. The estimated subject-specific shape model are shown in Fig. 6.3 (d). This two-step shape learning method can also be used in hand modeling.

6.5 Discussion

There are two issues to be discussed in this chapter. First, while our simple shape model has significantly reduced the number of involving Gaussians to achieve fast computation, our articulated shape model could be a little coarse for some complex

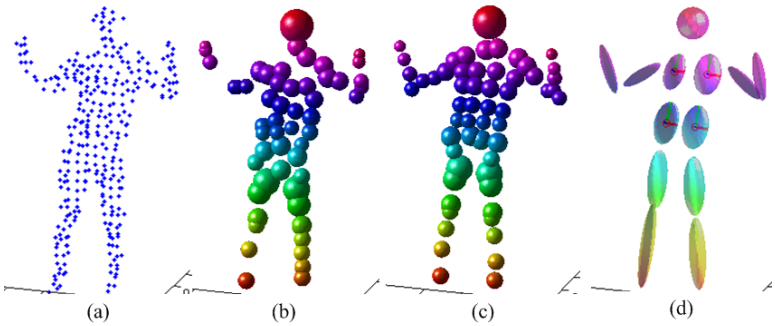


Figure 6.3: Subject-specific shape estimation. (a) Observation, (b) Estimated SoG model without LLE topology constraint. (c) Estimated SoG model with the LLE topology constraint. (d) Final multivariate SoG model mapped from (c).

poses or serious self-occlusions. In those situation, a detailed shape model is required for higher accuracy and robustness. Second, instead of directly learning the shape parameters of the multivariate Gaussian kernels, we use an auxiliary univariate SoG model and estimate the shape parameters in a two-step approach. While this method is easy to implement, the estimated shape parameters could not be as accurate as using the direct estimation method. Without considering the efficiency, we could also use some other global optimization methods to learn the subject-specific model, like Particle Swarm Optimization (PSO), which could be easier and more robust to estimate the parameters in multivariate Gaussian kernels. These two issues will guide our future work.

CHAPTER 7

POSE TRACKING BY ARTICULATED GKC

7.1 Introduction

Articulated human/hand pose tracking is one of the fundamental research topics in the field of computer vision and machine learning due to their wide applications and related technologies, such as Human Computer Interaction (HCI), Robotics, Computer Animation and Biomechanics. Recently, the launch of low-cost RGB-D sensors (e.g., Kinect) has further triggered a large amount of research due to the additional depth information and easy foreground/background segmentation. In this research, we propose an efficient and robust sequential pose tracking algorithm by introducing three constraints (visibility, continuity and self-intersection) which is successfully applied to pose tracking of both body and hand from a single depth sensor. In this chapter, we first develop a Octree-based method to represent the point cloud data as a collection of univariate Gaussian kernels. Then, we introduce our objective function, followed by its gradient-based optimization. Moreover, we develop a failure detection and recovery strategy to encourage robust and smooth pose tracking.

Our algorithm is simple and efficient and can run at about 10 FPS on a i7 desktop PC without GPU acceleration. We evaluate our articulated pose tracking algorithm on two depth benchmark datasets, i.e., (body) [24] and (hand) [76], which shows that the accuracy of pose estimation is competitive compared to the best results reported so far [26, 25, 28].

7.2 Point Cloud Representation

In the framework of kernel correlation-based registration, both the template and the observation are represented by a sum (combination) of kernels. In the previous chapter, we have built a simple yet effective shape model represented by multivariate Gaussian kernels. Here, we aim to convert the raw point cloud data into a set of Gaussian kernels, by which the similarity between our template and the observation can be directly measured by our derived AGKC, which is defined in (6.5) and (5.5). The simplest way is just down sampling the original point cloud data and assign a Gaussian kernel with an identical variance at each point. However, large amounts of noise and outlier of the raw data will be involved, leading to poor pose tracking results. In this research, our idea is to cluster the 3D points into many small pieces and each cluster of points can be approximated by a isotropic Gaussian so that the observed point cloud is robustly represented by a SoG-based model. Inspired by the Quad-tree which aims to cluster the image pixels with a similar color in [72], we novelly exploit an Octree to directly partition the point cloud in the 3D space.

An Octree is a tree data structure in which each internal node has exactly eight children. It is a useful shape representation tool to partition a 3D space by recursively subdividing it into eight octants. We illustrate the comparison of Quad-tree and Octree for partitioning in Fig. 7.1.

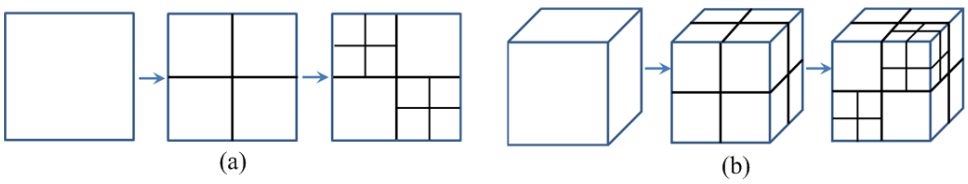


Figure 7.1: (a) Quad-tree partition in 2D. (b) Octree partition in 3D.

Here, we develop our own partition metric, i.e., if points in a Octree node has a large standard deviation along the depth direction (greater than a threshold η_{depth}), we divide the node into eight sub-nodes, up to a maximum Octree level n_{level}. Then,

points in each leaf node cube (illustrated as adjacent points in the same color in Fig. 7.2 (b)) are represented by an isotropic (univariate) Gaussian G_j centered at the mean of the points with the variance σ_j^2 that is set to be the square of half-length of a side of the cube. Consequently, we obtain a compact and noise-reduced univariate SoG representation $\mathcal{K}_B$ of a point cloud as shown in Fig. 7.2 (c). It is noted that the number of Gaussian kernels in the observation is obviously smaller than the number of points in the raw data, which indicates the computational complexity is reduced significantly. Next, we will exploit the AGKC defined in (6.5) and (5.5) to build our objective function and its optimization for pose tracking.

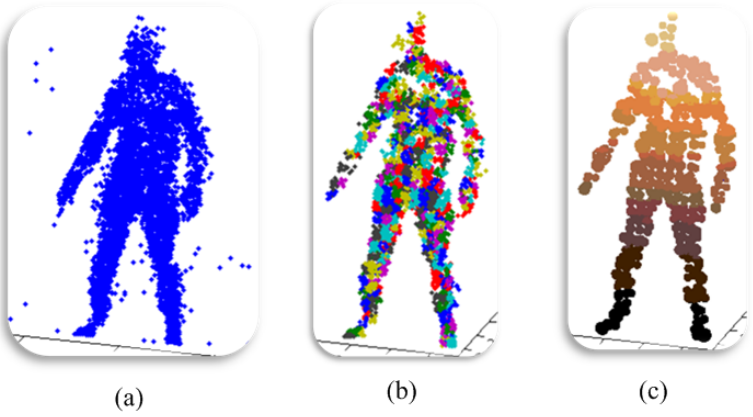


Figure 7.2: An illustration of a SoG-based representation of point cloud data. (a) the raw point cloud. (b) the partition results (adjacent points in the same color have similar depth). (c) The observation represented by a sum of isotropic Gaussian kernels.

7.3 Objective Function

The goal of the pose tracking algorithm is to estimate the pose parameters $\mathbf{\Theta}$ at time t from an observed point cloud by minimizing an objective function and utilizing previous pose information. The system framework includes two parts, i.e., initialization for shape modeling which has been introduced in previous chapter and pose

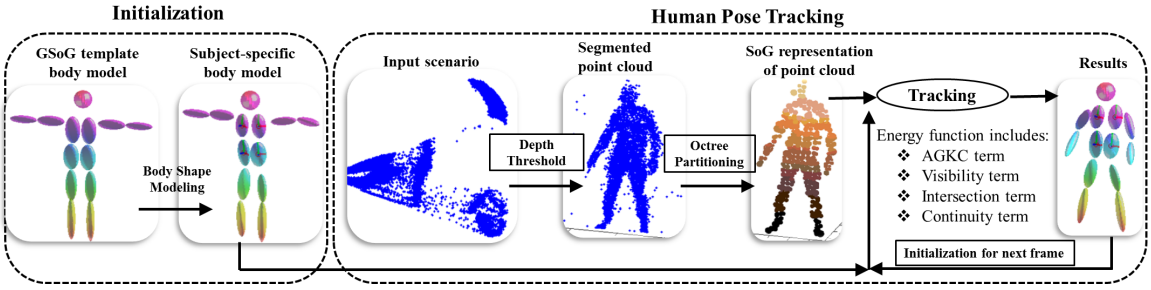


Figure 7.3: We estimate a SoG-bsaed subject-specific body model during initialization. Given a new frame for tracking, we first segment the target by converting the depth map into a point cloud that is further represented by a SoG using Octree. Then, the body model is fitted into the observation by minimizing the given objective function to estimate the underlying articulated pose parameters.

tracking, as shown in Fig. 7.3. We define our objective function that includes the similarity term that is AGKC $sMKC(\widetilde{\mathcal{K}_A}(\mathbf{\Theta}), \mathcal{K}_B)$ defined in (6.5) and (5.5), and three additional constraints. The first is a visibility detection term Vis to cope with the incomplete data problem from self-occlusion; The second one is an intersection penalty $E_{int}(\mathbf{\Theta})$ to discourage the intersection of two body segments; The third one is a continuity term $E_{con}(\mathbf{\Theta})$ to enforce a smooth pose transition during sequential tracking. Then pose estimation is formulated as an optimization problem with the following objective function:

$$\hat{\mathbf{\Theta}} = \arg\min_{\mathbf{\Theta}} \Bigg\{ -\sum_{l=1}^{L} \frac{1}{\omega_l} \sum_{k=1}^{K_l} \sum_{j=1}^{N} MKC(\widetilde{\boldsymbol{\mu}}_{l,k}^{(A)}(\mathbf{\Theta}), \boldsymbol{\mu}_j^{(B)}) \cdot Vis(l,k) + \eta E_{int}(\mathbf{\Theta}) + \gamma E_{con}(\mathbf{\Theta}) \Bigg\}, \tag{7.1}$$

where the first term is the negative of $sMKC$ in (6.5); $E_{int}(\mathbf{\Theta})$ and $E_{con}(\mathbf{\Theta})$ are the intersection and continuity term respectively; λ, γ are the weights to balance the last two terms, and $Vis(l,k)$ is the visibility of the k_{th} Gaussian in the segment S_l, defined

as,

$$Vis(l,k) = \begin{cases} 0 \text{ if the Gaussian is invisible,} \\ 1 \qquad\qquad \text{otherwise.} \end{cases} \tag{7.2}$$

In the following, we will introduce each term in details.

7.3.1 Similarity Term

The most important part of our objective function is the similarity term that is just the kernel correlation function defined in (6.5) and (5.5). As mentioned before, we address the pose tracking problem by maximizing the similarity between the template and the observation through the AGKC. It is noted that maximizing the kernel correlation function is equivalent to minimizing its negative. Consequently, we have the similarity term as,

$$E_{sim} = -\sum_{l=1}^{L} \frac{1}{\omega_l} \sum_{k=1}^{K_l} \sum_{j=1}^{N} MKC(\widetilde{\boldsymbol{\mu}}_{l,k}^{(A)}(\boldsymbol{\Theta}), \boldsymbol{\mu}_j^{(B)}), \tag{7.3}$$

where K_l is the number of Gaussian kernels in the segment S_l (totally we have L segments with the equality $K_1 + \cdots + K_l + \cdots + K_L = M$), and $\frac{1}{\omega_l}$ means the weight of the corresponding segment S_l. More details can be found in Chapter 6.3.

7.3.2 Visibility Detection Term

Due to the monocular depth sensor configuration, there exists self-occlusion problem, shown as an example in Fig. 7.4 (a), where the body turned around almost 90 degree and only half of the body can be seen. Obviously, the full body template model can not match well with the incomplete point cloud. To address the incomplete data problem like Fig. 7.4 (a), we develop a simple visibility detection term to identify and exclude the invisible Gaussian kernels from the subject shape model.

Our idea is that a large overlap among multiple Gaussians in the projected image plane may indicate an occlusion. To compute the overlap area analytically, we again use the auxiliary univariate SoG (the one used in the first-step shape learning in

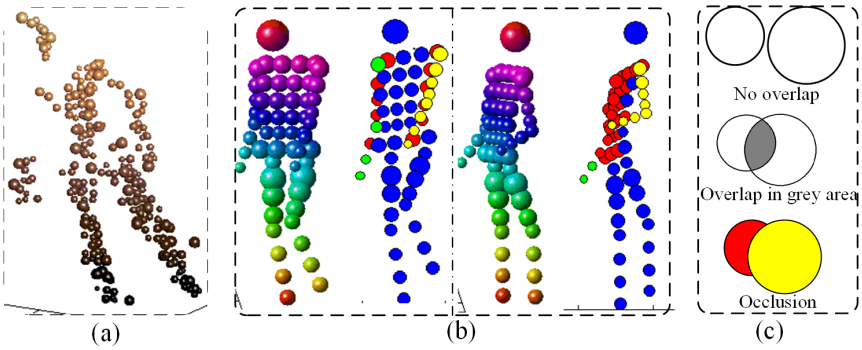


Figure 7.4: (a) Incomplete point cloud. (b) Two examples of auxiliary SoG body models and their orthographic projections, where the red circles denote the occluded components, and the yellow and green ones are remained. (c) Overlaps on the 2D projection plane.

Chapter 6.4) for occlusion handling. Similar to [27], we use the pose in previous frame to deform the template and compute the projected overlap area under an assumption that previous pose should close to the current one. First, each Gaussian of the template model under the previous pose is orthographically projected to the 2D image plane along the depth direction, resulting in a set of circles whose radii are set to be the square root of the corresponding variances. Then, we compute the overlap area between every two circles. As shown in Fig. 7.4 (c), if the overlap area of any pairwise circles is larger than a percentage ϵ (e.g. $\epsilon = \frac{1}{3}$) of the area of the smaller one, we declare an occlusion. The Gaussian kernel which is closer to the camera is remained, otherwise, it is occluded. Then, we map the auxiliary SoG model to the multivariate SoG model with the pre-defined mapping, which has been used for shape modeling in Chapter 6.4. Finally, we count the number of occluded circles in each body segment to decide its visibility. If 3 of 4 Gaussian kernels are invisible, the corresponding segment is excluded during optimization.

It is worth mentioning that the visibility detection will be triggered only when the body is not face to the camera and has turned around a large relative angle with

camera, e.g. ±50 degree. The relative angle between the body and the camera is the estimated pitch angle of the torso (we assume the body exactly faces to the camera at the initialization).

7.3.3 Intersection Penalty Term

In previous SoG-based methods [73, 74, 29], to avoid the situation that two or more body segments intersect with each other so that a Gaussian in the observation makes multiple contributions to the similarity measure, an artificial *clamping function* was used to constrain the similarity of each Gaussian kernel in $\mathcal{K}_B$,

$$E_{sim}(\mathbf{\Theta}) = \sum_{j \in \mathcal{K}_B} \min \left(\Big(\sum_{i \in \mathcal{K}_A} E_{ij}(\mathbf{\Theta}) \Big), \omega E_{jj} \right), \tag{7.4}$$

where E_{ij} is the similarity between the i_{th} Gaussian in $\mathcal{K}_A$ and the j_{th} Gaussian in $\mathcal{K}_B$; E_{jj} is the similarity of a Gaussian with itself in $\mathcal{K}_B$, ω is a constant to weight E_{jj}. More details can be found in [72, 30]. However, this clamping operation introduces some discontinuity so that the objective function is not differentiable everywhere, which may hinder the performance of the gradient-based optimizer. In this research, we develop an intersection penalty term to replace the artificial clamping function which is naturally deduced from the proposed GKC framework in equation (6.4). The idea is that two separated body segments in $\mathcal{K}_A$ are treated as a template $\mathcal{K}_{s1}$ and a target $\mathcal{K}_{s2}$, and then their KC can be used to measure the degree of their intersection,

$$E'_{int}(\mathbf{\Theta}) = MKC(\widetilde{\mathcal{K}_{s1}}(\mathbf{\Theta}), \widetilde{\mathcal{K}_{s2}}(\mathbf{\Theta})). \tag{7.5}$$

When two segments intersect each other, their KC becomes large, resulting a larger intersection penalty. In practice, we consider five self-intersection cases, i.e., head-torso, forearm-arm, upper limb-torso, shank-thigh and lower limb-torso, as shown in Fig. 7.5. $E_{int}(\mathbf{\Theta})$ which is the sum of KC measures of the five cases can be considered as a soft constraint which preserves the continuity and differentiability of the objective

function.

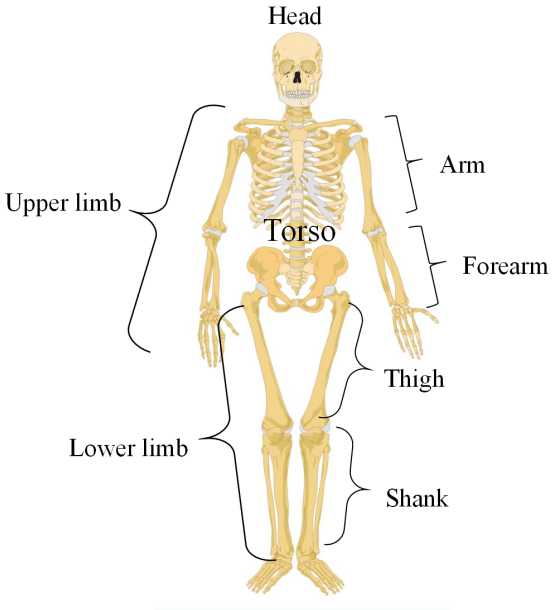


Figure 7.5: The illustration and definition of body segments in the self-intersection term.

7.3.4 Continuity Term

To encourage smooth sequential tracking, we augment the objective function with a continuity term as follows,

$$E_{con}(\mathbf{\Theta}^{(t)}) = \sum_{d=1}^{D} \left[\left(\mathbf{\Theta}_d^{(t)} - \mathbf{\Theta}_d^{(t-1)} \right) - \left(\mathbf{\Theta}_d^{(t-1)} - \mathbf{\Theta}_d^{(t-2)} \right) \right]^2, \tag{7.6}$$

where $\mathbf{\Theta}^{(t)}$ is the present pose and $\mathbf{\Theta}^{(t-1)}, \mathbf{\Theta}^{(t-2)}$ are the previous two poses; d represents the dimension index in $\mathbf{\Theta}$. The continuity term penalizes the current pose to have a large deviation from previous frames, ensuring relatively smooth and continuous pose estimation. Since the objective is continuous and differentiable, we can use the efficient gradient-based optimization methods to estimate the pose parameters. In the next section, we will provide more details on the optimization and the derivative of the objective function.

7.4 Gradient-based Optimization

Due to the differentiable AGKC function and the computational benefits of quaternion-based rotation representation, we can explicitly derive the derivative of the objective function E with respect to $\mathbf{\Theta}$ and employ a gradient-based optimizer. Different with a variant of steepest descent used in [72, 73], we employ a Quasi-Newton method (L-BFGS [120]) because of its faster convergence. For simplicity of notification, we ignore the visibility detection term in (7.1) without changing the derivatives (the visibility detection term can not be ignored in the implement). We has the following form:

$$\begin{aligned}\frac{\partial E(\mathbf{\Theta})}{\partial \mathbf{\Theta}} &= -\frac{\partial sMKC(\widetilde{\mathcal{K}_A}(\mathbf{\Theta}), \mathcal{K}_B)}{\partial \mathbf{\Theta}} \\ &\quad +\lambda \frac{\partial E_{int}(\mathbf{\Theta})}{\partial \mathbf{\Theta}} + \gamma \frac{\partial E_{con}(\mathbf{\Theta})}{\partial \mathbf{\Theta}} \\ &= -\sum_{l=1}^{L} \frac{1}{\omega_l} \sum_{k=1}^{K_l} \sum_{j=1}^{N} \frac{MKC(\widetilde{\boldsymbol{\mu}}_{l,k}^{(A)}(\mathbf{\Theta}), \boldsymbol{\mu}_j^{(B)})}{\partial \mathbf{\Theta}} \\ &\quad +\lambda \frac{\partial E_{int}(\mathbf{\Theta})}{\partial \mathbf{\Theta}} + \gamma \frac{\partial E_{con}(\mathbf{\Theta})}{\partial \mathbf{\Theta}}. \end{aligned} \tag{7.7}$$

We denote $\mathbf{r} = [r_1, r_2, r_3, r_4]^T$ as an un-normalized quaternion, which is normalized to $\mathbf{p} = [x, y, z, w]^T$ according to $\mathbf{p} = \frac{\mathbf{r}}{\|\mathbf{r}\|}$. We represent the pose $\mathbf{\Theta}$ as $[\mathbf{t}, \mathbf{r}^{(1)}, \ldots, \mathbf{r}^{(L)}]$, where $\mathbf{t} = [t_1, t_2, t_3] \in \mathbb{R}^3$ defines a global translation, L is the number of joints to be estimated, and each normalized quaternion $\mathbf{p}^{(l)}$ from $\mathbf{r}^{(l)} \in \mathbb{R}^4$ defines the relative rotation of the l_{th} joint. Defined in (5.5), $\boldsymbol{\mu}_{l,k} = [a, b, c]^T$ is the center of k_{th} Gaussian kernel in the segment S_l which is transformed from its local coordinate $\widetilde{\boldsymbol{\mu}}_{l,k}$ through transformation T_l in (6.2) and the corresponding covariance matrix $\Sigma_{l,k}$ is approximated and updated from the previous pose under an assumption that is adjacent poses should be close to each other. We explicitly represent every pairwise kernel correlation using equation (5.5) and take derivative with respect to each pose

parameter,

$$\frac{\partial MKC}{\partial t_n} = \frac{\partial MKC}{\partial \boldsymbol{\mu}_{l,k}} \frac{\partial \boldsymbol{\mu}_{l,k}}{\partial t_n}, \quad (n = 1, 2, 3) \tag{7.8}$$

$$\frac{\partial MKC}{\partial r_m^{(l)}} = \frac{\partial MKC}{\partial \boldsymbol{\mu}_{l,k}} \frac{\partial \boldsymbol{\mu}_{l,k}}{\partial r_m^{(l)}}, \ (m = 1, \cdots, 4) \tag{7.9}$$

Then, every derivative of the pairwise kernel correlation will be sum over to obtain the gradient vector. In the following, we explicitly write the derivative in terms of each pose parameter. First, we consider $\frac{\partial MKC(\widetilde{\boldsymbol{\mu}}_i(\boldsymbol{\Theta}), \boldsymbol{\mu}_j)}{\partial \mathbf{t}}$, where $\widetilde{\boldsymbol{\mu}}_i(\boldsymbol{\Theta}) = \boldsymbol{\mu}_i$.

According to the chain rule, we have,

$$\frac{\partial MKC_{ij}}{\partial t_1} = \frac{\partial MKC_{ij}}{\partial \boldsymbol{\mu}_i} \cdot \frac{\partial \boldsymbol{\mu}_i}{\partial t_1} \tag{7.10}$$

$$= \begin{bmatrix} \frac{\partial MKC_{ij}}{\partial a} & \frac{\partial MKC_{ij}}{\partial b} & \frac{\partial MKC_{ij}}{\partial c} \end{bmatrix} \begin{bmatrix} \frac{\partial a}{\partial t_1} \\ \frac{\partial b}{\partial t_1} \\ \frac{\partial c}{\partial t_1} \end{bmatrix}, \tag{7.11}$$

Since the covariance matrixes Σ_1 and Σ_2 in equation (5.5) are the symmetric matrix, we can obtain:

$$\frac{\partial MKC_{ij}}{\partial a} = -MKC(\widetilde{\boldsymbol{\mu}}_i(\boldsymbol{\Theta}), \boldsymbol{\mu}_j)) \cdot (\widetilde{\boldsymbol{\mu}}_i(\boldsymbol{\Theta}) - \boldsymbol{\mu}_j)(\Sigma_i + \Sigma_j)^{-1}[1 \ \ 0 \ \ 0]^T, \tag{7.12}$$

$$\frac{\partial MKC_{ij}}{\partial b} = -MKC(\widetilde{\boldsymbol{\mu}}_i(\boldsymbol{\Theta}), \boldsymbol{\mu}_j)) \cdot (\widetilde{\boldsymbol{\mu}}_i(\boldsymbol{\Theta}) - \boldsymbol{\mu}_j)(\Sigma_i + \Sigma_j)^{-1}[0 \ \ 1 \ \ 0]^T, \tag{7.13}$$

$$\frac{\partial MKC_{ij}}{\partial c} = -MKC(\widetilde{\boldsymbol{\mu}}_i(\boldsymbol{\Theta}), \boldsymbol{\mu}_j)) \cdot (\widetilde{\boldsymbol{\mu}}_i(\boldsymbol{\Theta}) - \boldsymbol{\mu}_j)(\Sigma_i + \Sigma_j)^{-1}[0 \ \ 0 \ \ 1]^T, \tag{7.14}$$

Also, we can derive

$$\begin{bmatrix} \frac{\partial a}{\partial t_1} \\ \frac{\partial b}{\partial t_1} \\ \frac{\partial c}{\partial t_1} \end{bmatrix} = \frac{\partial (\mathbf{R} \cdot \widetilde{\boldsymbol{\mu}}_i + \mathbf{t})}{\partial t_1}$$

$$= [1 \ \ 0 \ \ 0]^T, \tag{7.15}$$

where $\mathbf{R}$ is the rotation that transfer the $\widetilde{\boldsymbol{\mu}}_i$ at its local coordinate to the global coordinate. Consequently, we have $\frac{\partial MKC_{ij}}{\partial t_1} = \frac{\partial MKC_{ij}}{\partial a}$, which can be calculated in

equation (7.12). In the same way, we can derive $\frac{\partial MKC_{ij}}{\partial t_2} = \frac{\partial MKC_{ij}}{\partial b}$ and $\frac{\partial MKC_{ij}}{\partial t_3} = \frac{\partial MKC_{ij}}{\partial c}$, which can be calculated in (7.13) and (7.14), respectively.

Next, we consider the gradient on the non-normalized quaternion $\frac{\partial MKC_{ij}}{\partial \boldsymbol{r}}$. Similarly, using the chain rule, we have:

$$\frac{\partial MKC_{ij}}{\partial r_1} = \frac{\partial MKC_{ij}}{\partial \boldsymbol{\mu}_i} \cdot \frac{\partial \boldsymbol{\mu}_i}{\partial r_1} \tag{7.16}$$

$$= \begin{bmatrix} \frac{\partial MKC_{ij}}{\partial a} & \frac{\partial MKC_{ij}}{\partial b} & \frac{\partial MKC_{ij}}{\partial c} \end{bmatrix} \begin{bmatrix} \frac{\partial a}{\partial r_1} \\ \frac{\partial b}{\partial r_1} \\ \frac{\partial c}{\partial r_1} \end{bmatrix}, \tag{7.17}$$

where $\frac{\partial MKC_{ij}}{\partial a}$, $\frac{\partial MKC_{ij}}{\partial b}$ and $\frac{\partial MKC_{ij}}{\partial c}$ have been derived in (7.12), (7.13) and (7.14). We know that the center of a Gaussian at T-pose in the local coordinate $\widetilde{\boldsymbol{\mu}}_i$ is converted to the global (world) coordinate through a series of rotation matrixes. Here, we use $\mathbf{R}$ to represent those rotation matrixes for simplicity. Therefore, we have:

$$\begin{bmatrix} \frac{\partial a}{\partial r_1} \\ \frac{\partial b}{\partial r_1} \\ \frac{\partial c}{\partial r_1} \end{bmatrix} = \frac{\partial \mathbf{R}}{\partial \mathbf{p}} \frac{\partial \mathbf{p}}{\partial r_1} \cdot \widetilde{\boldsymbol{\mu}}_i{}^T$$

$$= \begin{bmatrix} \frac{\partial \mathbf{R}}{\partial x} & \frac{\partial \mathbf{R}}{\partial y} & \frac{\partial \mathbf{R}}{\partial z} & \frac{\partial \mathbf{R}}{\partial w} \end{bmatrix} \begin{bmatrix} \frac{\partial x}{\partial r_1} \\ \frac{\partial y}{\partial r_1} \\ \frac{\partial z}{\partial r_1} \\ \frac{\partial w}{\partial r_1} \end{bmatrix} \cdot \widetilde{\boldsymbol{\mu}}_i{}^T, \tag{7.18}$$

where $\mathbf{p} = [x, y, z, w]^T$ is the normalized quaternion according to $\mathbf{p} = \frac{\mathbf{r}}{\|\mathbf{r}\|}$. Since the conversion formulation between rotation matrix and quaternion is given, it is straightforward to calculate $\frac{\partial \mathbf{R}}{\partial x}$, $\frac{\partial \mathbf{R}}{\partial y}$, $\frac{\partial \mathbf{R}}{\partial z}$, $\frac{\partial \mathbf{R}}{\partial w}$. For example,

$$\frac{\partial \mathbf{R}}{\partial x} = \begin{pmatrix} 0 & 2y & 2z \\ 2y & -4x & -2w \\ 2z & 2w & -4x \end{pmatrix}, \tag{7.19}$$

In general, we have the derivative of the normalization terms $\frac{\partial p_m}{\partial r_j}$ as,

$$\frac{\partial p_m}{\partial r_j} = \frac{\delta_{mj}}{\|\mathbf{r}\|} - \frac{r_m r_j}{\|\mathbf{r}\|^{\frac{3}{2}}}, \tag{7.20}$$

where $\delta_{mj} = 1$ when $m = j$, otherwise, $\delta_{mj} = 0$.

As a result, we can calculate $\frac{\partial MKC_{ij}}{\partial r_1}, \cdots, \frac{\partial MKC_{ij}}{\partial r_4}$ for one non-normalized quaternion. It is similar to derive the other set of quaternion. Only one difference is the definition of the rotation matrix $\mathbf{R}$ in (7.18), where $\mathbf{R}$ is constructed by the father rotation matrix multiplying the current rotation matrix.

Since $E_{int}(\mathbf{\Theta})$ is naturally deduced from the GKC, the derivative of $E_{int}(\mathbf{\Theta})$ can also be calculated by a similar way according to (7.8), (7.9). Since $E_{con}(\mathbf{\Theta}^{(t)})$ in (7.6) is a standard quadratic form, we have its gradient expression directly as:

$$\frac{\partial E_{con}(\mathbf{\Theta}^{(t)})}{\partial \mathbf{\Theta}_d^{(t)}} = 2\left[\left(\mathbf{\Theta}_d^{(t)} - \mathbf{\Theta}_d^{(t-1)}\right) - \left(\mathbf{\Theta}_d^{(t-1)} - \mathbf{\Theta}_d^{(t-2)}\right)\right], \tag{7.21}$$

where $d = 1, \ldots, D$ is the index in the pose parameter vector.

After obtaining the derivative of each pairwise GKC, we sum up over the derivatives of all the pairwise in each dimensionality of $\mathbf{\Theta}$ to compose the pose vector. By now, we have prepared well all the required derivative expression for the gradient-based optimization.

In the Quasi-Newton optimization, the initialization of $\mathbf{\Theta}^{(t)}$ in each frame is the estimated pose in the previous frame and the pose in the first frame is assumed to be close to a standard T-pose facing to the camera, similar to the treatment in many other algorithms.

7.5 Failure Detection and Recovery

Although gradient-based local optimization is effective in most cases, it is still possible to be stuck at local minima and cannot be recovered automatically, especially when there is a dramatic fast articulated pose change or significant self-occlusion. To cope with this problem, we incorporate Particle Swarm Optimization (PSO) with gradient-based search to balance the effectiveness and efficiency when exploring the high-dimensional parameter space [121, 122, 123]. To reduce the computational load, some data-driven detectors will be helpful to provide a good initialization and narrow the search space. In [76], some finger detectors are used to effectively combine gradient-based ICP and sampling-based PSO for real-time articulated hand tracking. Similar ideas can be incorporated in our tracking framework where Gaussian KC-based optimization is treated as the local optimizer and PSO is used for global search. Additional detectors are necessary to support real-time performance of the hybrid global-local optimization which are beyond the scope of this research.

The hybrid optimization with PSO and AGKC is only necessary when a tracking failure is detected. We evaluate the average KC for all N univariate Gaussian kernels in the observation ($\mathcal{K}_B$) by checking the following condition:

$$\frac{1}{N} sMKC(\widetilde{\mathcal{K}_A}(\mathbf{\Theta}), \mathcal{K}_B) < \eta_{fail}, \tag{7.22}$$

where $sMKC(\cdot)$ is defined in (6.5) and η_{fail} is a threshold. When (7.22) is true, it indicates that a number of Gaussian kernels in $\mathcal{K}_B$ are not aligned or explained by the deformed shape template $\mathcal{K}_A$. Then the local-global optimization scheme will be triggered for failure recovery, where PSO is involved to allow the global PSO sampling along with the local gradient-based AGKC optimization.

7.6 Experimental results

In this section, we will evaluate our articulated pose tracking algorithm on two benchmark datasets, i.e., human body [24] [1] and hand [76] [2], both of which are captured by a single depth sensor. We will validate the effectiveness of different constrains and compare with state-of-the-art approaches quantitatively and qualitatively. Also, we will comprehensively analysis the algorithm efficiency and the failure detection and recovery strategy.

7.6.1 Experimental Setup

Testing Database: We first use the depth benchmark dataset SMMC-10 [24] to evaluate our algorithm for human pose tracking and compare it with a series of state-of-the-art methods. The SMMC-10 dataset consists of 28 depth sequences, which include various human motion types. The ground truth data are the 3D marker positions which are recorded by an optical tracker. The significant noise and outliers in this depth dataset makes it challenging yet proper for evaluating algorithm robustness and accuracy. Secondly, we also use the benchmark dataset in [76] to test our algorithm for articulated hand tracking. This dataset is reported as one of the most challenging ones due to the fast hand motion and considerable self-occlusion. Performance evaluation on the first dataset is both quantitative and qualitative to validate the efficacy and efficiency of our algorithm for human pose tracking, while that of the second one is mainly qualitative to demonstrate the potential of the proposed framework for a different articulated structure.

Evaluation Metrics: We adopt two metrics for performance evaluation of human pose estimation. One evaluation metric is to directly measure the averaged error of the Euclidean distance between the ground-truth markers and estimated ones over

[1] Available at: http://ai.stanford.edu/ varung/

[2] Available at: http://research.microsoft.com/en-us/um/people/yichenw/handtracking/index.html

all markers across all frames,

$$\bar{e} = \frac{1}{N_f}\frac{1}{N_m}\sum_{k=1}^{N_f}\sum_{i=1}^{N_m}\|\mathbf{p}_{ki} - \mathbf{v}_i^{disp} - \hat{\mathbf{p}_{ki}}\|, \tag{7.23}$$

where N_f and N_m are the number of frames and markers, respectively; $\mathbf{p}_{ki}$ and $\hat{\mathbf{p}_{ki}}$ are the ground-truth location of the i_{th} marker and the estimated one in the k_{th} frame, respectively; $\mathbf{v}_i^{disp}$ is the displacement vector of the i_{th} marker. Because the marker definitions across different body models are different, the inherent and constant displacement $\mathbf{v}^{disp}$ should be subtracted from the error, as a routine in most methods. In this paper, we manually chose 40 frames with ground truth in the #6 Sequence for the calculation of $\mathbf{v}^{disp}$. To make $\mathbf{v}^{disp}$ independent at any pose, we project each markers on the centerline of its corresponding segment and compute an offset $\mathbf{v}^{disp}$ in the local coordinate system for each segment individually. The other evaluation metric is the percentage of correctly estimated joints whose Euclidean distance errors are less than $10cm$.

Algorithm Parameters: Some empirical parameters we used for human pose tracking throughout our experiments are listed in Table 7.1. In Octree partitioning, the threshold η_{depth} and maximum Octree level n_{level} are set to be $20mm$ and 6, respectively. The weights η and γ in the objective function (7.1), and weight λ for LLE-based topology constrain in shape modeling (6.7) are set to be 0.001, 0.2 and 0.05, respectively. The threshold in failure detection (7.22) is set to be 9.

7.6.2 Effectiveness of the Constrains

To exhibit the effect of each regularization term introduced in the objective function, we conduct five experiments on the SMMC-10 dataset, where the continuity, visibility detection and intersection penalty terms as well as the subject-specific shape model are incorporated successively. Their corresponding tracking errors are shown in Fig. 7.6 (a), which shows that the tracking accuracy gradually improves with the

Table 7.1: Parameter settings and their description

Parameter	Description	Value
η_{depth}	a threshold to subdivide a Octree node	20 mm
n_{level}	maximum Octree level	6
η	a weight for self-intersection penalty in the objective function (7.1)	0.001
γ	a weight for continuity term in the objective function (7.1)	0.2
λ	a weight for LLE-based topology constrain in the shape modeling (6.7)	0.05
ϵ	a percentage to determine the occlusion in the visibility detection	$\frac{1}{3}$
η_{fail}	a threshold in failure detection (7.22)	9

addition of each of the three terms as well as the subject-specific shape model. Especially, in Sequences 24-27 where the occlusion problem is serious, the visibility and intersection terms make a significant contribution. It is also interesting to find that the continuity term has a a slight negative effect in Sequence 25 (Karate) due to its too strong penalty on the fast motion. However, the other terms and the shape model are able to improve the accuracy. Fig. 7.6 (b) and (c) illustrate the tracking error of the left elbow in Sequence 24 and that of the left knee in Sequence 27 respectively. It is clear that using additional terms (in red) achieves much smaller errors than the case without them (in blue). We visually compare the effect of the additional terms in Fig. 7.7, where it is observed that the results using additional terms (in green) are more accurate.

7.6.3 Accuracy Comparison

In Fig. 7.8 and Fig. 7.9, our algorithm is evaluated against the state-of-the-art methods in terms of two metrics. Failure recovery is only needed for Sequence 24, 25 and

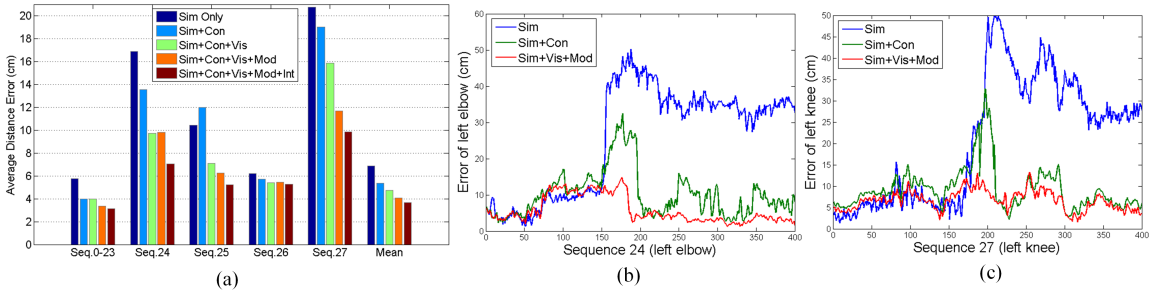


Figure 7.6: The effect of different terms in pose tracking. "Sim", "Con", "Vis", "Int" and "Mod" denote the *kernel correlation*, *continuity*, *visibility*, *intersection penalty* terms and the *subject-specific model*, respectively. (a) The improvements over different sequences. (b) The improvement over the left elbow in Sequence 24. (c) The improvement over the left knee in Sequence 27.

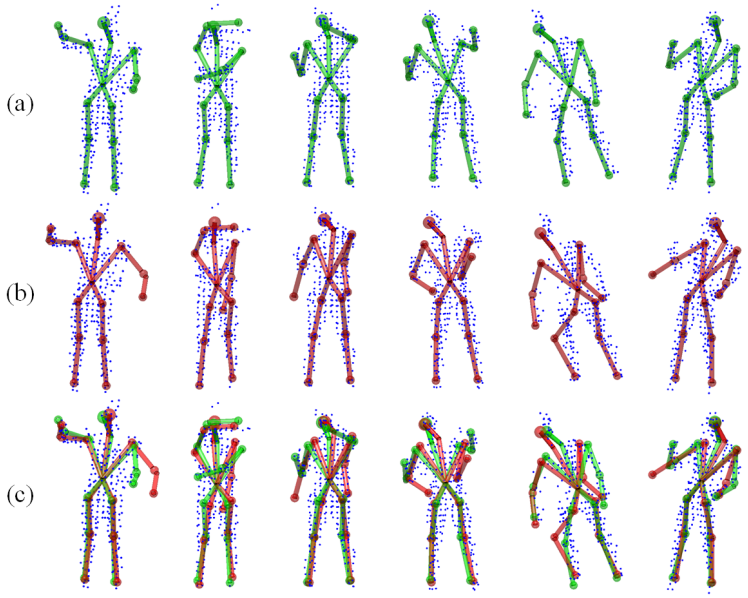


Figure 7.7: The visual comparison of the effect of the additional terms. (a) Results with the additional terms and subject-specific model. (b) Results only with the kernel correlation. (c) Two results are merged together for comparison (the one from (a) in green and the one from (b) in red).

27, and our approach achieves the average error $3.56cm$ on the SMMC-10 dataset and it is close to the best results so far (around $3.4 \sim 3.6cm$) [25, 26, 28] where a

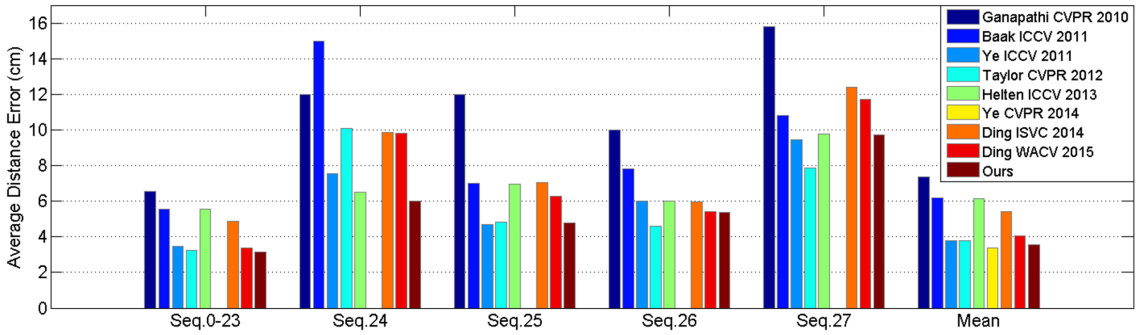


Figure 7.8: The accuracy comparison with the state-of-the-art methods, i.e., Ganapathi et al. [24], Baak et al. [21], Ye et al. [25], Taylor et al. [26], Helten et al. [27], Ye et al. [28], Ding et al. [29] and Ding et al. [30] in distance error (cm). Except our previous works [29, 30] and this research, all the others use both a large scale database and a mesh model or either of them. Since no individual result of each sequence is reported in [28], we only show its average result.

database or a detailed mesh model or both are involved. If no failure detection and recovery are involved with real-time performance for all sequences, the average error is $3.71cm$. Moreover, we notice that our results are better than the original SoG algorithm (reported in [29]) and [27] where additional inertial sensors were used. It also outperforms our early GSoG method [30], which is mainly due to the proposed segment-scaled AGKC and the differentiable intersection penalty term. Compared with most other methods, our algorithm is simpler with lower complexity. Furthermore, we compare the precision of joint estimation (Metric II) in Fig. 7.9. It shows that our algorithm is still comparable with the best algorithms [28, 31].

7.6.4 Efficiency Analysis

In all generative methods for pose estimation, the computational complexity is expressed as $O(MN)$, where M is the number of vertices in a surface model and N is the number of points in the observation point set. Due to the multivariate SoG body shape representation and Octree-based point cloud representation, M and N in our

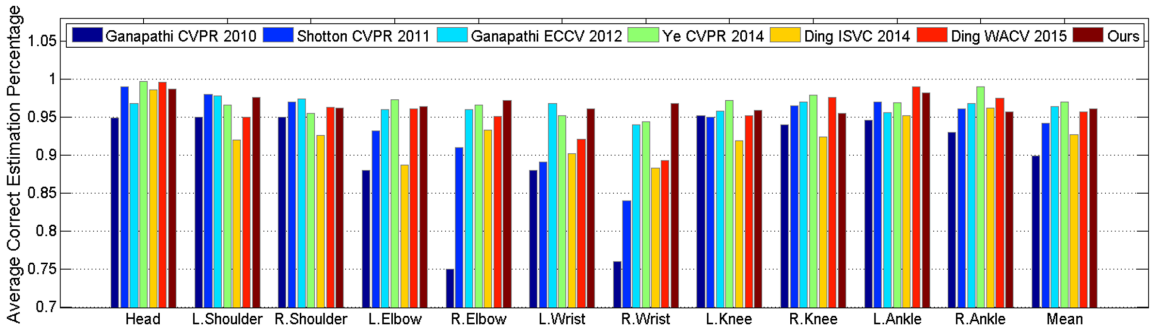


Figure 7.9: The precision comparison with the state-of-the-art methods, i.e., Ganapathi et al. [24], Shotten et al. [20], Ganapathi et al. [31], Ye et al. [28], Ding et al. [29] and Ding et al. [30].

approach are much less than those in most methods and M in the multivariate SoG is only about a quarter of that in the standard SoG-based shape model, leading to a low computational cost. We implement our tracking algorithm in C++ with the L-BFGS optimization library [124]. Currently, the efficiency is evaluated on a PC without GPU acceleration. We allow maximum 30 iterations in the first frame (similar to a standard T-pose) and then 15 iterations in the following frames, and we ignore the computation time of background segmentation using a depth threshold and the Octree partitioning which is very efficient. We can achieve about $10 \sim 15$ frames per second without the code optimization for human pose tracking. If the hybrid local-global optimizer is employed in three sequences (#24, 25, 27), the computational cost is increased due to PSO-based failure recovery, leading to a lower frame rate. In this work, we used 10 particles and 20 generations in the PSO-assisted local-global optimizer to test the effectiveness of the failure detection and recovery. However, it is possible to keep the real-time performance if our algorithm can be integrated with some data-driven detectors as those used in [76] to initialize and reduce the search space. Due to the collective nature of AGKC and PSO, our pose tracking algorithm (with failure recovery) is compatible with GPU-based parallel computing for further acceleration.

7.6.5 Failure Detection and Recovery

We track the average AGKC value in each frame according to (7.22) to detect a failure. The average AGKC in some exemplary sequences are shown in Fig. 7.10. We can observe that the values of average AGKC in most sequences are relatively smooth and higher than the threshold η_{fail}, which indicates that no failures are detected. On the other hand, the values of average AGKC in Sequence 24 and 27 dramatically decrease at some frames and are lower than the threshold η_{fail}, which implies there exist tracking failures.

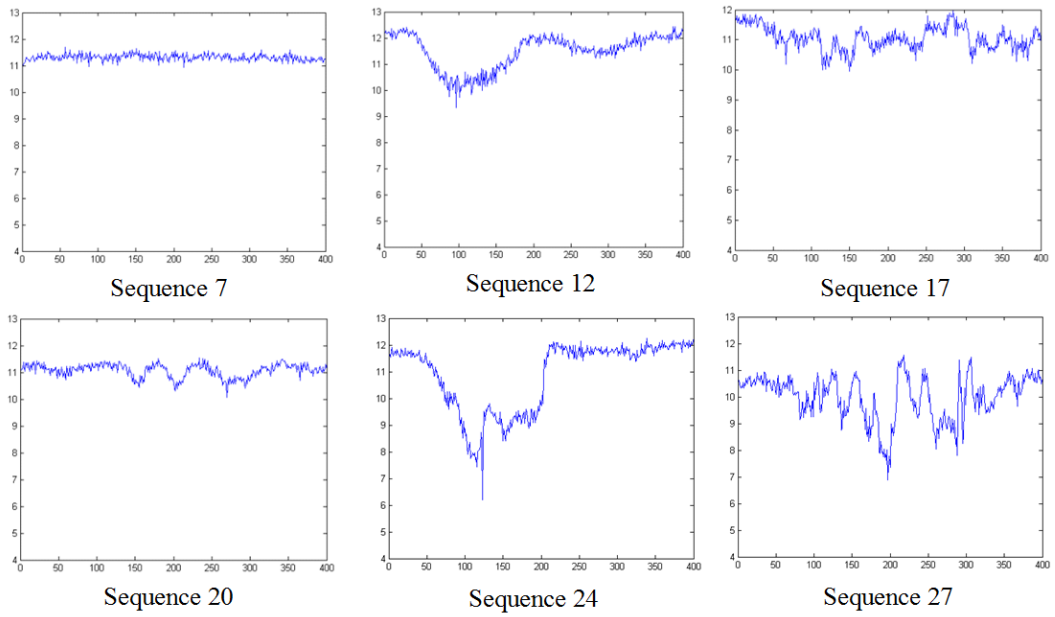


Figure 7.10: The average AGKC in some exemplary sequences.

In our experiment, only three SMMC-10 sequences (#24, #25 and #27) has a couple of detected failures. However, most hand sequences require failure recovery due to fast motion change and complex self-occlusion. Fig. 7.11 shows the average AGKC with/without the failure recovery in Sequence #25 of SMMC-10 and Sequence #1 of hand motion. As shown in Fig. 7.11 (a) and (b), pose estimation fails from frame #174, where its average AGKC value drops below the threshold ($\eta_{fail} = 9$). Then, the recovery is triggered in the following frames, until the average AGKC value

becomes larger than η_{fail}. Without failure recovery, the pose tracker could be trapped in local minima in the following frames, as shown in the red curve in Fig 7.11 (b). On the other hand, Fig 7.11 (c) visualizes the recovered pose estimation result in frame 200. The similar results for a hand sequence are shown in Fig 7.11 (d,e,f), where the failure is detected in frame 74 and a good recovery is obtained at frame 97.

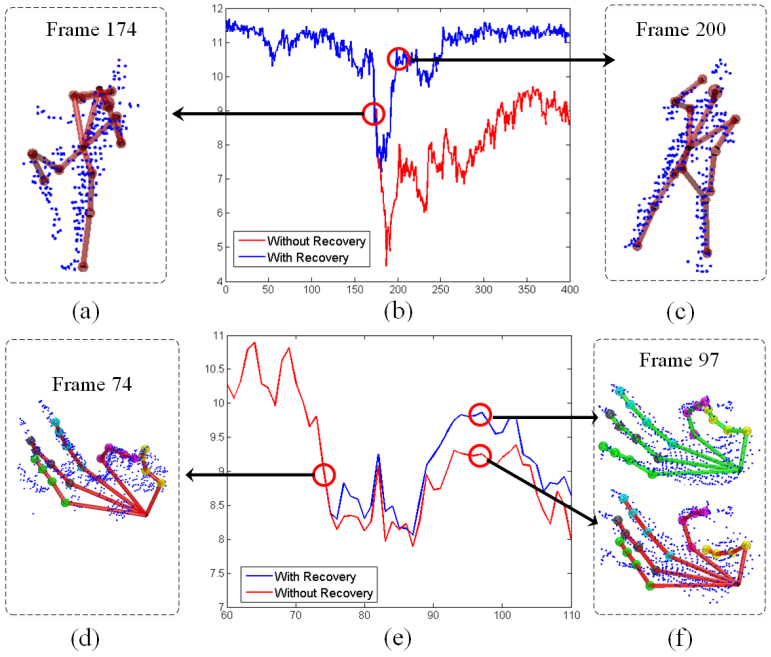


Figure 7.11: Two illustrations of failure detection and recovery in the human and hand motion. (a) and (d) The human/hand pose tracking failures are detected. (b) and (e) The values of average KC with (blue line) and without (red line) the failure recovery. (c) and (f) The recovered human pose in frame 200, and the comparison of hand poses (with/without recovery) in frame 97.

While most tracking failures can be successfully recovered for full-body pose tracking, the current hybrid optimization strategy is still not ready to handle complicated hand motion yet. The main reason is that AGKC has too many local minima in hand tracking, which deuterates when there are fast articulated pose changes and complex self-occlusion problems. A more advanced failure detector [125] could be helpful to

reduce false alarms. More importantly, some finger detectors similar to that used in [76] could mitigate this problem by reducing the search space and providing a better initialization.

7.7 Discussion

Some pose estimation results of SMMC-10 sequences are shown in Fig. 7.14. While the estimated poses are accurate in most frames for all sequences, and the failure recovery is only triggered in a couple of frames in three sequences, our tracker may still fail in a few frames of some sequences, as shown in the last row of Fig. 7.14. We also evaluate our algorithm on several sequences from the hand dataset and compare with the ground truth qualitatively in Fig. 7.12. Since the hand motion is rapidly changing and highly articulated, there exists significant self-occlusion in most hand sequences. Failure detection and recovery are required for most hand sequences. Although the hybrid optimizer shows promising results in our experiments, it may still fail in some frames of highly complex articulated motion. Some hand tracking failures are shown in Fig. 7.13.

There are two possible reasons which will guide our future research. First, the visibility term in the objective function may not be accurate since it is determined from the previous frame and used an approximate orthographic projection, especially in the case of fast motion or changing camera view. We could address this by incorporating the predicted pose into the visibility term or employing some other powerful visibility detection techniques. Second, there are still many local minima in the objective function mainly due to the self-occlusion problems, and a better optimizer is needed to take advantage of the differentiability of AGKC. PSO is effectively but costly, and it must be confined to a small search space. Integrating additional pose detector or other bottom-up features could improve initialization and narrow the search space which are the two main keys to efficient and effective optimization in articulated pose

tracking of the full-body and hands.

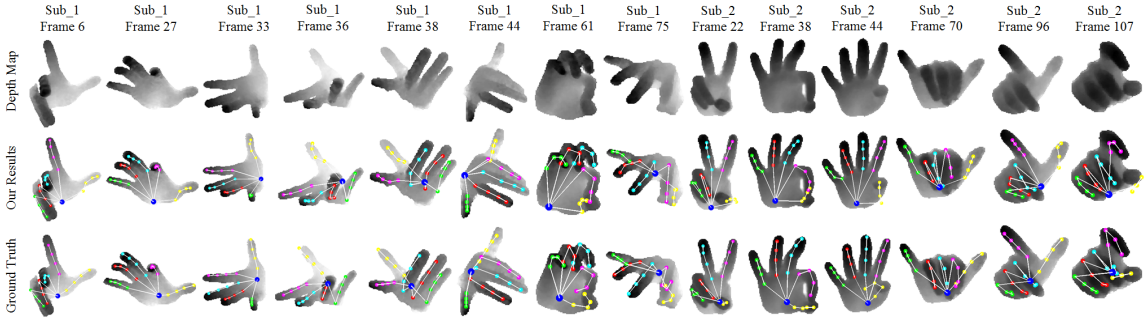


Figure 7.12: The illustrations of some articulated hand tracking results.

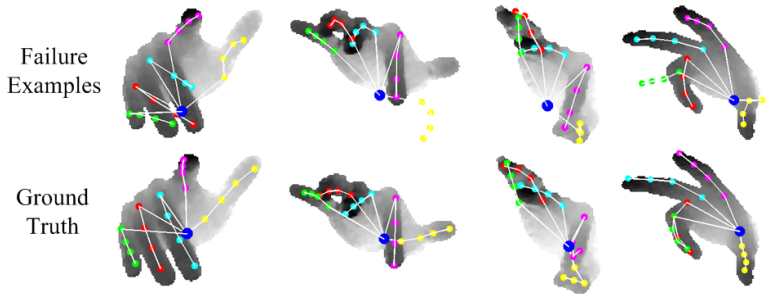


Figure 7.13: Examples of hand pose tracking failure.

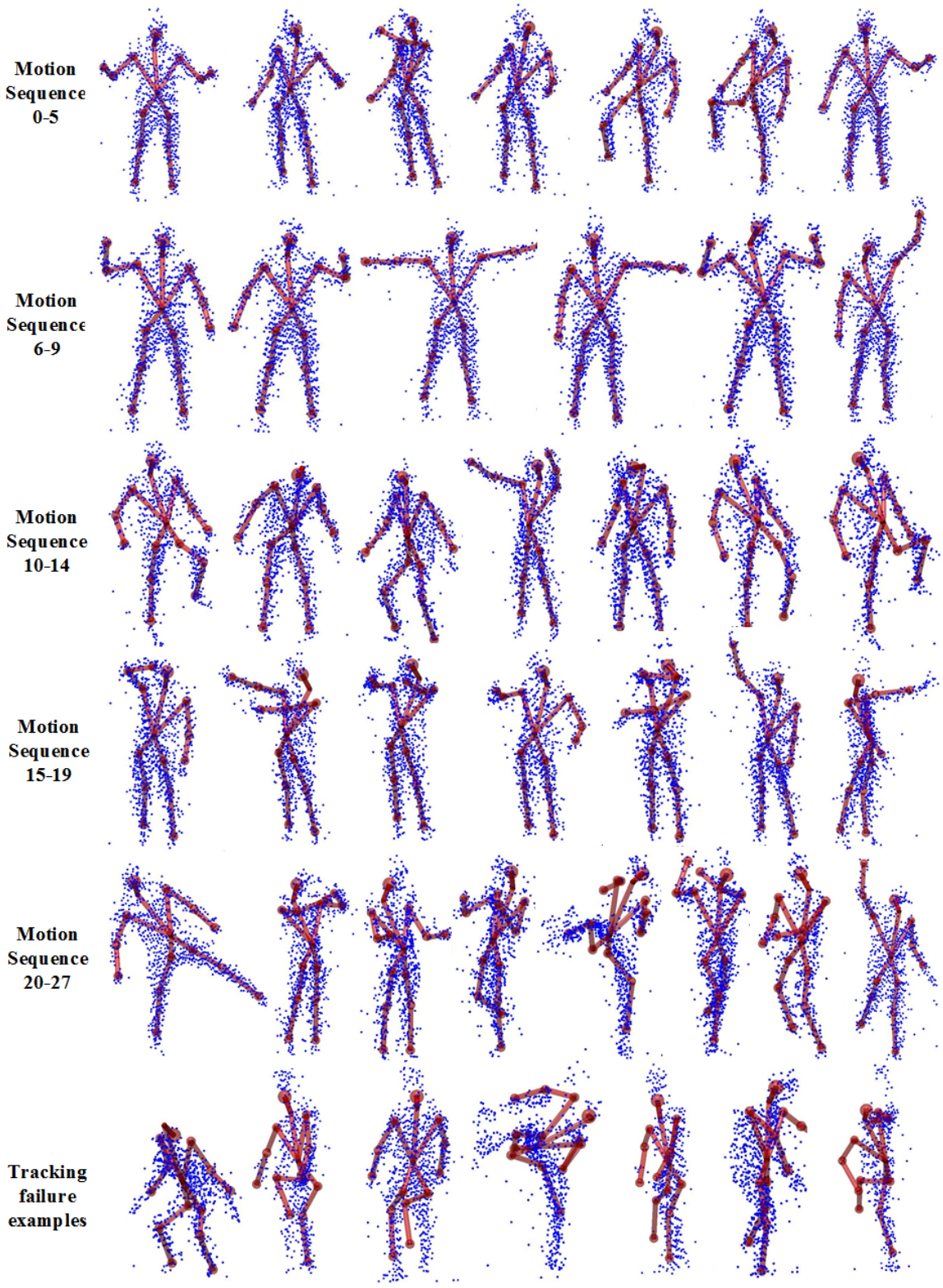


Figure 7.14: The illustrations of some human pose tracking results and some tracking failure examples from all motion sequences.

CHAPTER 8

CONCLUSIONS AND FUTURE RESEARCH

8.1 Conclusions

This dissertation presented a series of approaches for human motion analysis from three perspectives, i.e., manifold learning-based gait motion modeling, articulated body shape representation and efficient articulated pose estimation. Firstly, we have presented a new JGPM learning algorithm that is able to jointly optimize the gait and pose variables simultaneously for gait modeling. Compared with the original JGPM which is learned by a two-step learning process, a more straightforward one-step GPLVM-based learning algorithm is developed. Also, since less hyper-parameters are involved, the computational complexity is considerably reduced, which makes it possible for large scale learning. Experimental results show that our proposed JGPM has the superior performance for motion interpolation and filtering compared with the existing GPLVM-based algorithms as well as the original JGPM-I and JGPM-II and it is comparable with JGPM-III in the numerical results.

Also, we have proposed a multi-layer JGPM in order to enhance the representability and flexibility of the single layer JGPM for more complex gait motion modeling. There are two key techniques to make the multi-layer JGPM computational feasible, i.e., *training data diversification* and *topology-aware local learning*. The first technique is simple yet effective to generate a rich set of simulated training motion with different walking styles, which allows us to learn a more powerful model without increasing the size of the original training data. This data diversification technique naturally supports a multi-layer toroidal or cylindrical structure as the topological

prior for manifold learning. The second technique enables that the model learning can be implemented efficiently and effectively on a larger training dataset and the resultant manifold is compliant with the topological prior both locally and globally. We demonstrate the effectiveness of our approach by synthesizing the high-quality motions from the multi-layer model. The experimental results show that the multi-layer JGPM outperforms several existing GPLVM-based models in terms of motion interpolation, reconstruction and filtering.

On the other hand, to achieve human pose estimation from a single depth sensor, we have developed a novel generative method, i.e., articulated Gaussian kernel correlation (AGKC)-based shape model for pose tracking. First, We have extended the Gaussian kernel correlation from the univariate Gaussian to the multivariate one and developed a generalized Gaussian KC (GKC) framework that provides a continuous and differentiable similarity measure between a template and an observation, both of which are represented by a collection of univariate and/or multivariate Gaussians. Second, to accommodate articulated body deformation, we embed a quaternion-based articulated skeleton into a multivariate SoG-based shape model and further develop an AGKC function to measure the similarity between the template and the observation. Consequently, articulated pose parameters are estimated by maximizing AGKC under three additional constraints, i.e., visibility, intersection penalty and continuity. A simple yet effective failure detection and recovery strategy has been implemented to enhance the robustness and smoothness of pose tracking. Also, the new AGKC function naturally supports a subject-specific shape modeling with a LLE-based topology constraint. We have evaluated our proposed tracker on two public depth datasets, and the experimental results are encouraging and promising compared with the state-of-the-art algorithms, especially considering its simplicity and efficiency. Our algorithm can achieve fast and accurate human pose estimation with competitive accuracy and precision, and the proposed GKC and AGKC functions can also be applied to other

articulated structures, like hand.

8.2 Future Research

Our future research will focus on the two issues, i.e., accurately shape modeling and advanced articulated pose tracking. We will discuss each of them in detail in the following.

- *How to develop a more accurate shape modeling algorithm?*

  In current research, we use a general template which is composed by 13 multivariate Gaussian kernels to represent the body shape and we have developed a subject-specific shape modeling algorithm to capture the variances between different subjects. While it is effective and efficient for pose tracking in most of our experiments, it may be limited to handle a subject with loosely fit clothing or with significant articulated deformation. The main reason is our rough shape model cannot handle the complicated non-rigid deformation on the surface of the human body. Also, the shape model is too coarse to accurately determine which body segment is visible or not. We believe that a more detailed and parametric body model is still needed to handle those challenging problems.

  In our future research, we will employ a more detailed body shape which is represented by many small univariate/multivariate Gaussian kernels, by which the subject-specific body shape can be depicted more accurately using our shape modeling algorithm. Also, we will construct a topology-based relationship among the Gaussian kernels on each body segment via a parametric mapping function, so that the Gaussian kernels on each body segment will not share the same transformation, but have their own ones. Specifically, we want to exploit an additional soft (non-rigid) transformation on each body segment. During pose tracking, we hope to estimate the transformation between two adjacent

Gaussians that are on the same body segment. In this way, the pose parameters and the frame-specific shape deformation can be optimized simultaneously. There are mainly two challenges. First, since hundreds of additional parameters are involved, a high performance optimization method is necessary for effective pose and shape estimation. Second, the computational complexity will dramatically increase. Some efficient approximation methods and GPU acceleration could be required to sustain the efficiency. Both challenges will guide our future research.

- *How to develop a more accurate, robust and efficient pose tracking algorithm?*

  In current research, while we enhance the similarity measure between the template and an observation with three additional constraints, and we also have a simple failure detection and recovery strategy, our tracker may still fail in some frames, especially for the complex hand motion. The first reason is the visibility detection term may not be accurate since it is determined from the previous frame and used an approximate orthographic projection. Second, there are still many local minima in the objective function mainly due to the self-occlusion and self-intersection problems. To solve the problem of visibility detection, we will incorporate a prediction strategy to provide a pose that is more close to the current one for the visibility detector. Also, replacing the current projection-based visibility detection, we will employ some other powerful visibility detection techniques in the future, such as the Hidden Point Removal in [126]. To solve the problem of poor local minima, a better optimizer is required to take advantage of our differentiable AGKC function. The local-global optimization strategy is effective but costly, and it must be confined to a small search space. In our future research, we will integrate an additional pose detector or other bottom-up data-driven methods to improve the initialization and narrow down the search space. Moreover, we will explore a more advanced failure detector

to adapt to different application environment, like the technique used in [125].

To further improve the accuracy and robustness of the Mocap system, we could employ the inertial sensors which can provide accurate rotation angle information to complement the limitation of monocular depth sensor-based methods, especially for self-occlusion handing and pose estimation from the side-view observation. In our future research, a sensor fusion framework could be developed to recover more accurate and robust pose estimation from the results of a depth sensor and inertial sensors.

BIBLIOGRAPHY

[1] "People Tracking for Visual Surveillance." http://www.siebel-research.de/objects/ tracked3large.jpg.

[2] "Visual Surveillance." http://www.inext.uts.edu.au/images/program3.jpg.

[3] Y. Abe, C. K. Liu, and Z. Popovic, "Momentum-based parameterization of dynamic character motion," *Graphical Models*, vol. 68, no. 2, pp. 194 – 211, 2006.

[4] Y. Li, T. Wang, and H. Shum, "Motion texture: a two-level statistical model for character motion synthesis," *ACM Transactions on Graphics*, vol. 21, pp. 465–472, July 2002.

[5] "Biomechanics." http://www.bath.ac.uk/health/sportsandexercise/shesimages.jpg.

[6] "Humanoid Robot." http://t1.gstatic.com/images.

[7] "Kinect Sensor for HCI." http://www.ubergizmo.com/wp-content/Kinect-Poker.jpg.

[8] "Kinect Sensor for Xbox 360." http://imsdn.sec.smsft.com/dynimg/IC584396.png.

[9] "Kinect Sensor Version 2." http://blogs.msdn.com/b/kinectforwindows/archive/2014/03/27/revealing-kinect-for-windows-v2-hardware.aspx.

[10] "Rehabilitation." http://www.allina.com/ahs/owatonna.jpg.

[11] "Gait Identification." http://gtresearchnews.gatech.edu/images/gait3.jpg.

[12] "Fall-risk Assessment." http://t3.gstatic.com/images.

[13] "Body Controller in Video Game." http://static.gamespot.com/uploads/scale-super/1179/11799911/2385603-fallonnbc.jpg.

[14] "Rehabilitation Solution for Stroke Victims." http://blogs.msdn.com/resized-image.ashx/size/520x309/key/communityserver-blogs-components-weblogfiles /00-00-01-49-02/7752.jintronix_2D00_kinect_2D00_stroke_2D00_recovery.jpg.

[15] "Virtual Clothes Fitting." http://i.ytimg.com/vi/1jbvnk1T4vQ/ maxresde-fault.jpg.

[16] "Marker Based Mocap System." http://www.fvrcgi.it/Documenti/ vol-ume12CamStand.jpg.

[17] "Mocap and Animation Demo." https://upload.wikimedia.org/wikipedia /commons/thumb/6/6d/Activemarker2.PNG/300px-Activemarker2.PNG.

[18] "MVN BIOMECH." http://www.xsens.com/en/movement-science.

[19] "Xsens." https://www.xsens.com/.

[20] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kip-man, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[21] A. Baak, M. Muller, G. Bharaj, H.-P. Seidel, and C. Theobalt, "A data-driven approach for real-time full body pose reconstruction from a depth camera," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2011.

[22] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[23] X. Zhang and G. Fan, "Joint gait-pose manifold for video-based human motion estimation," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshop on Machine Learning for Vision-based Motion Analysis*, 2011.

[24] V. Ganapathi, C. Plagemann, D. Koller, and S. Thrun, "Real time motion capture using a single time-of-flight camera," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[25] M. Ye, X. Wang, R. Yang, L. Ren, and M. Pollefeys, "Accurate 3D pose estimation from a single depth image," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2011.

[26] J. Taylor, J. Shotton, T. Sharp, and A. Fitzgibbon, "The vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[27] T. Helten, M. Muller, H.-P. Seidel, and C. Theobalt, "Real-time body tracking with one depth camera and inertial sensors," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2013.

[28] M. Ye and R. Yang, "Real-time simultaneous pose and shape estimation for articulated objects with a single depth camera," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[29] M. Ding and G. Fan, "Fast human pose tracking with a single depth sensor using sum of Gaussians models," in *Advances in Visual Computing*, vol. 8887 of *Lecture Notes in Computer Science*, pp. 599–608, 2014.

[30] M. Ding and G. Fan, "Generalized sum of Gaussians for real-time human pose tracking from a single depth sensor," in *Proc. IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 47–54, Jan 2015.

[31] V. Ganapathi, C. Plagemann, D. Koller, and S. Thrun, "Real-time human pose tracking from range data," in *Proc. European Conference on Computer Vision (ECCV)*, 2012.

[32] "Vicon Mocap System." http://www.vicon.com/.

[33] "OptiTrack Mocap System." http://www.optitrack.com/.

[34] "CMU Human Motion Capture Database." http://mocap.cs.cmu.edu.

[35] M. R. Menard, M. E. McBride, D. Sanderson, and D. D. Murray, "Comparative biomechanical analysis of energy-storing prosthetic feet," *Archives of Physical Medicine and Rehabilitation*, vol. 73, no. 5, pp. 451–8, 1992.

[36] R. Hannah, J. Morrison, and A. Chapman, "Kinematic symmetry of the lower limbs.," *Archives of Physical Medicine and Rehabilitation*, vol. 65, no. 4, p. 155, 1984.

[37] L. Chou, S. Song, and L. Draganich, "Predicting the kinematics and kinetics of gait based on the optimum trajectory of the swing limb," *Journal of biomechanics*, vol. 28, no. 4, pp. 377–385, 1995.

[38] S. E. Pierotti, R. A. Brand, R. H. Gabel, D. R. Pedersen, and W. R. Clarke, "Are leg electromyogram profiles symmetrical?," *Journal of orthopaedic research*, vol. 9, no. 5, pp. 720–729, 1991.

[39] A. Yao, J. Gall, L. V. Gool, and R. Urtasun, "Learning probabilistic non-linear latent variable models for tracking complex activities," in *Proc. Neural Information Processing Systems (NIPS)*, 2011.

[40] X. Lan and D. Huttenlocher, "A unified spatio-temporal articulated model for tracking," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[41] L. Sigal, S. Bhatia, S. Roth, M. Black, and M. Isard, "Tracking loose-limbed people," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[42] L. Sigal and M. Black, "Measure locally, reason globally: Occlusion-sensitive articulated pose estimation," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[43] B. Rosenhahn, C. Schmaltz, T. Brox, J. Weickert, D. Cremers, and H. P. Seidel, "Markerless motion capture of man-machine interaction," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[44] M. Vondrak, L. Sigal, and O. Jenkins, "Physical simulation for probabilistic motion tracking," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[45] M. A. Brubaker, D. Fleet, and A. Hertzmann, "Physics-based person tracking using simplified lower-body dynamics," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[46] M. A. Brubaker and D. Fleet, "The kneed walker for human pose tracking," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[47] M. A. Brubaker, D. J. Fleet, and A. Hertzmann, "Physics-based person tracking using the anthropomorphic walker," *International Journal of Comput Vision*, vol. 87, no. 1-2, pp. 140–155, 2010.

[48] J. B. Tenenbaum, V. d. Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[49] A. Elgammal and C.-S. Lee, "Separating style and content on a nonlinear manifold," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[50] C.-S. Lee and A. Elgammal, "Modeling view and posture manifolds for tracking," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2007.

[51] A. Elgammal and C.-S. Lee, "Tracking people on a torus," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 3, pp. 520–538, 2009.

[52] N. Lawrence, "Probabilistic non-linear principal component analysis with Gaussian process latent variable models," *Journal of Machine Learning Research*, vol. 6, pp. 1783–1816, 2005.

[53] N. Lawrence and J. Q. Candela, "Local distance preservation in the GPLVM through back constraints," in *Proc. International Conference on Machine Learning (ICML)*, 2006.

[54] J. Wang, D. Fleet, and A. Hertzmann, "Gaussian process dynamical models for human motion," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 30, pp. 283–298, 2008.

[55] R. Urtasun, D. Fleet, A. Hertzmann, and P. Fua, "Priors for people tracking from small training sets," in *Proc. International Conference on Computer Vision (ICCV)*, 2005.

[56] R. Urtasum, D. Fleet, and P. Fua, "3D people tracking with Gaussian process dynamical models," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[57] J. Wang, D. Fleet, and A. Hertzmann, "Multifactor Gaussian process models for style-content separation," in *Proc. International Conference on Machine Learning (ICML)*, 2007.

[58] R. Urtasun, D. Fleet, A. Geiger, J. Popovic, T. Darrel, and N. Lawrence, "Topologically-contraint latent variable models," in *Proc. International Conference on Machine Learning (ICML)*, 2008.

[59] X. Zhang and G. Fan, "Dual gait generative models for human motion estimation from a single camera," *IEEE Trans. on System, Man, and Cybernetics*, vol. 40, pp. 1034–1049, 2010.

[60] M. Ding, G. Fan, X. Zhang, S. Ge, and L.-S. Chou, "Structure-guided manifold learning for video-based motion estimation," in *Proc. IEEE International Conference on Image Processing*, 2012.

[61] C. Plagemann, V. Ganapathi, D. Koller, and S. Thrun, "Real-time identification and localization of body parts from depth images," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2010.

[62] R. Girshick, J. Shotton, P. Kohli, A. Criminisi, and A. Fitzgibbon, "Efficient regression of general-activity human poses from depth images," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2011.

[63] D. Tang, T.-H. Yu, and T.-K. Kim, "Real-time articulated hand pose estimation using semi-supervised transductive regression forests," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, pp. 3224–3231, IEEE, 2013.

[64] S. R. Fanello, C. Keskin, S. Izadi, P. Kohli, D. Kim, D. Sweeney, A. Criminisi, J. Shotton, S. B. Kang, and T. Paek, "Learning to be a depth camera for close-range human capture and interaction," *ACM Transactions on Graphics*, vol. 33, pp. 86:1–86:11, July 2014.

[65] P. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, Feb 1992.

[66] S. Pellegrini, K. Schindler, and D. Nardi, "A generalisation of the ICP algorithm for articulated bodies," in *Proc. British Machine Vision Conference (BMVC)*, 2008.

[67] D. Hahnel, S. Thrun, and W. Burgard, "An extension of the icp algorithm for modeling nonrigid objects with mobile robots," in *Proc. International Joint Conference on Artificial Intelligence (IJCAI)*, 2003.

[68] J. Gall, A. Fossati, and L. Van Gool, "Functional categorization of objects using real-time markerless motion capture," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1969–1976, June 2011.

[69] I. Oikonomidis, N. Kyriazis, and A. A. Argyros, "Efficient model-based 3D tracking of hand articulations using kinect.," in *Proc. British Machine Vision Conference (BMVC)*, vol. 1, p. 3, 2011.

[70] 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 *Proc. IEEE International Conference on Computer Vision (ICCV)*, pp. 2088–2095, 2011.

[71] I. Oikonomidis, N. Kyriazis, and A. A. Argyros, "Tracking the articulated motion of two strongly interacting hands," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1862–1869, June 2012.

[72] C. Stoll, N. Hasler, J. Gall, H. Seidel, and C. Theobalt, "Fast articulated motion tracking using a sums of Gaussians body model," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2011.

[73] D. Kurmankhojayev, N. Hasler, and C. Theobalt, "Monocular pose capture with a depth camera using a Sums-of-Gaussians body model," in *Pattern Recognition*, vol. 8142 of *Lecture Notes in Computer Science*, pp. 415–424, 2013.

[74] S. Sridhar, A. Oulasvirta, and C. Theobalt, "Interactive markerless articulated hand motion tracking using RGB and depth data," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, pp. 2456–2463, Dec 2013.

[75] S. Sridhar, H. Rhodin, H. Seidel, A. Oulasvirta, and C. Theobalt, "Real-time hand tracking using a sum of anisotropic Gaussians model," in *Proc. International Conference on 3D Vision (3DV)*, pp. 319–326, Dec 2014.

[76] C. Qian, X. Sun, Y. Wei, X. Tang, and J. Sun, "Realtime and robust hand tracking from depth," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[77] T. Helten, A. Baak, G. Bharaj, M. Muller, H.-P. Seidel, and C. Theobalt, "Personalization and evaluation of a real-time depth-based full body tracker," in *Proc. International Conference on 3D Vision (3DV)*, June 2013.

[78] X. Wei, P. Zhang, and J. Chai, "Accurate realtime full-body motion capture using a single depth camera," *ACM Transactions on Graphics*, vol. 31, Nov. 2012.

[79] A. M. Peter and A. Rangarajan, "Maximum likelihood wavelet density estimation with applications to image and shape matching," *IEEE Transactions on Image Processing*, vol. 17, no. 4, pp. 458–468, 2008.

[80] Y. Wang, K. Woods, and M. McClain, "Information-theoretic matching of two point sets," *IEEE Transactions on Image Processing*, vol. 11, no. 8, pp. 868–872, 2002.

[81] A. Myronenko and X. Song, "Point set registration: Coherent point drift," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 2262–2275, Dec 2010.

[82] R. Horaud, F. Forbes, M. Yguel, G. Dewaele, and J. Zhang, "Rigid and articulated point registration with expectation conditional maximization," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 33, pp. 587–602, March 2011.

[83] B. Jian and B. C. Vemuri, "Robust point set registration using Gaussian mixture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1633–1645, 2011.

[84] Y. Tsin and T. Kanade, "A correlation-based approach to robust point set registration," in *Proc. European Conference on Computer Vision (ECCV)*, pp. 558–569, Springer, 2004.

[85] D. W. Scott and W. F. Szewczyk, "From kernels to mixtures," *Technometrics*, vol. 43, no. 3, pp. 323–335, 2001.

[86] Y. Tsin and T. Kanade, "A correlation-based model prior for stereo," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. I–135–I–142 Vol.1, June 2004.

[87] P. J. Huber, *Robust statistics*. Springer, 2011.

[88] 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 *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[89] L. Ballan, A. Taneja, J. Gall, L. V. Gool, and M. Pollefeys, "Motion capture of hands in action using discriminative salient points," in *Proc. European Conference on Computer Vision (ECCV)*, 2012.

[90] H. Sidenbladh, M. J. Black, and D. J. Fleet, "Stochastic tracking of 3D human figures using 2D image motion," in *Proc. European Conference on Computer Vision (ECCV)*, 2000.

[91] R. Horaud, M. Niskanen, G. Dewaele, and E. Boyer, "Human motion tracking by registering an articulated surface to 3D points and normals," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, pp. 158–163, Jan 2009.

[92] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: real-time tracking of the human body," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 780–785, Jul 1997.

[93] R. Plankers and P. Fua, "Articulated soft objects for multiview shape and motion capture," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 1182–1187, Sept 2003.

[94] T. B. Moeslund and E. Granum, "A survey of computer vision-based human motion capture," *Computer Vision and Image Understanding*, vol. 81, no. 3, pp. 231 – 268, 2001.

[95] T. B. Moeslund, A. Hilton, and V. Krger, "A survey of advances in vision-based human motion capture and analysis," *Computer Vision and Image Understanding*, vol. 104, no. 2?, pp. 90 – 126, 2006.

[96] R. Poppe, "Vision-based human motion analysis: An overview," *Computer Vision and Image Understanding*, vol. 108, Oct. 2007.

[97] G. Rogez, J. Rihan, J. Guerrero, and C. Orrite, "Monocular 3-D gait tracking in surveillance scenes," *Cybernetics, IEEE Transactions on*, vol. 44, pp. 894–909, June 2014.

[98] J. Martinez-del Rincon, M. Lewandowski, J.-C. Nebel, and D. Makris, "Generalized Laplacian eigenmaps for modeling and tracking human motions," *Cybernetics, IEEE Transactions on*, vol. 44, pp. 1646–1660, Sept 2014.

[99] H. Shum, E. Ho, Y. JIANG, and S. Takagi, "Real-time posture reconstruction for microsoft kinect," *Cybernetics, IEEE Transactions on*, vol. 43, pp. 1357–1369, Oct 2013.

[100] A. Elgammal and C.-S. Lee, "Inferring 3D body pose from silhouettes using activity manifold learning," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[101] C. Sminchisescu and A. Jepson, "Generative modeling for continuous non-linearly embedded visual inference," in *Proc. International Conference on Machine Learning (ICML)*, 2004.

[102] K. Grochow, S. L. Martin, A. Hertzmann, and Z. Popović, "Style-based inverse kinematics," in *Proc. ACM SIGGRAPH*, 2004.

[103] Y. Li, T. Wang, and H.-Y. Shum, "Motion texture: A two-level statistical model for character motion synthesis," *ACM Transactions on Graphics*, vol. 21, no. 3, pp. 465–472, 2002.

[104] J. Chai and J. K. Hodgins, "Performance animation from low-dimensional control signals," *ACM Transactions on Graphics*, vol. 24, no. 3, pp. 686–696, 2005.

[105] M. Tejera, D. Casas, and A. Hilton, "Animation control of surface motion capture," *Cybernetics, IEEE Transactions on*, vol. 43, pp. 1532–1545, Dec 2013.

[106] G. Fan, X. Zhang, and M. Ding, "Gaussian process for human motion modeling: A comparative study," in *Proc. IEEE Machine Learning for Signal Processing (MLSP)*, 2011.

[107] A. Elgammal and C.-S. Lee, "The role of manifold learning in human motion analysis," in *Human Motion*, vol. 36 of *Computational Imaging and Vision*, pp. 25–56, Springer Netherlands, 2008.

[108] M. Ding and G. Fan, "Multilayer joint gait-pose manifolds for human gait motion modeling," *IEEE Transactions on Cybernetics*, vol. PP, no. 99, pp. 1–1, 2014.

[109] M. Ding and G. Fan, "Multi-layer joint gait-pose manifold for human motion modeling," in *Proc. IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, 2013.

[110] N. D. Lawrence, "Learning for larger datasets with the Gaussian process latent variable model," in *Proc. International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2007.

[111] N. Lawrence, M. Seeger, and R. Herbrich, "Fast sparse Gaussian process methods: The informative vector machine," in *Proc. Neural Information Processing Systems (NIPS)*, 2003.

[112] J. Quiñonero Candela and C. E. Rasmussen, "A unifying view of sparse approximate Gaussian process regression," *Journal of Machine Learning Research*, vol. 6, pp. 1939–1959, 2005.

[113] E. Snelson and Z. Ghahramani, "Sparse Gaussian processes using pseudo-inputs," in *Proc. Neural Information Processing Systems (NIPS)*, 2006.

[114] N. D. Lawrence and A. J. Moore, "Hierarchical Gaussian process latent variable models," in *Proc. International Conference on Machine Learning (ICML)*, 2007.

[115] X. Zhang, G. Fan, and L. shan Chou, "Two-layer dual gait generative models for human motion estimation from a single camera," *Image and Vision Computing*, vol. 31, pp. 473 – 486, 2013.

[116] M. Law and A. Jain, "Incremental nonlinear dimensionality reduction by manifold learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 3, pp. 377–391, 2006.

[117] D. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 125–141, 2008.

[118] B. Schölkopf, K. Tsuda, and J.-P. Vert, *Kernel Methods in Computational Biology*. Computational Molecular Biology, Cambridge, MA, USA: MIT Press, Aug. 2004.

[119] M. P. Wand and M. C. Jones, *Kernel smoothing*. Chapman and Hall/CRC, 1994.

[120] D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Mathematical Programming*, vol. 45, pp. 503–528, Dec. 1989.

[121] J.-R. Zhang, J. Zhang, T.-M. Lok, and M. R. Lyu, "A hybrid particle swarm optimization-back-propagation algorithm for feedforward neural network training," *Applied Mathematics and Computation*, vol. 185, no. 2, pp. 1026 – 1037, 2007.

[122] S. Li, M. Tan, I. Tsang, and J.-Y. Kwok, "A hybrid PSO-BFGS strategy for global optimization of multimodal functions," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 41, pp. 1003–1014, Aug 2011.

[123] V. Plevris and M. Papadrakakis, "A hybrid particle swarm-gradient algorithm for global structural optimization," *Computer-Aided Civil and Infrastructure Engineering*, vol. 26, no. 1, pp. 48–68, 2011.

[124] libLBFGS, http://www.chokkan.org/software/liblbfgs/.

[125] S. L. Dockstader and N. S. Imennov, "Prediction for human motion tracking failures," *IEEE Transactions on Image Processing*, vol. 15, no. 2, pp. 411–421, 2006.

[126] S. Katz, A. Tal, and R. Basri, "Direct visibility of point sets," *ACM Transactions on Graphics*, vol. 26, July 2007.

VITA

Meng Ding

Candidate for the Degree of

Doctor of Philosophy

Dissertation: HUMAN MOTION ANALYSIS: FROM GAIT MODELING TO SHAPE REPRESENTATION AND POSE ESTIMATION

Major Field: Electrical Engineering

Biographical:

Education:

Completed the requirements for the Doctor of Philosophy in Electrical Engineering at Oklahoma State University, Stillwater, Oklahoma in July, 2015.

Completed the requirements for the Master of Science in Optoelectronic Engineering at Beijing Institute of Technology, Beijing, China in 2009.

Completed the requirements for the Bachelor of Science in Electrical Engineering at Beijing Institute of Technology, Beijing, China in 2007.

Experience:

Research Assistant, in VCIPL, Oklahoma State University, Aug.2010∼Present