CITY UNIVERSITY OF HONG KONG
香港城市大學

Modeling of Single Character Motions with Temporal Sparse Representation and Gaussian Processes for Human Motion Retrieval and Synthesis
基於時域稀疏表示和高斯過程的單角色動作模型的建立及其在動作檢索和生成的應用

Submitted to
Department of Computer Science
電腦科學系
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy
哲學博士學位

by

Zhou Liu Yang
周柳陽

August 2014
二零一四年八月

Abstract

3D motion capture (mocap) is the process to record and digitalize the movement of people or objects. Mocap technology is widely used in computer animation, man-machine interaction games, athletic training and 3D movies, etc. However, it is rather time and manpower consuming to capture human motions as it consists of calibration of the system and post processing of the captured artifacts. Therefore, it is essential to either reuse pre-captured data or develop effective methods to synthesize new motions. To reuse pre-captured data, we need an efficient retrieval mechanism to search for a particular motion from a large corpus. Human motion retrieval has proven to be challenging as human motion is high dimensional in both spatial and temporal domains. Besides, semantically similar motions are not necessarily numerically similar because of the speed variations. With the retrieved similar motions, we propose to synthesize human motion variations for intended applications. However, the joints of the human skeleton are highly correlated based on the articulated skeleton structure and it is challenging to synthesize natural human motions. In this thesis, we develop new methods to address the problem of reusing human motion capture data, which includes three sub-problems, i.e., *human motion retrieval*, *human motion variation*

synthesis and *human posture reconstruction*.

For *human motion retrieval*, an effective feature representation plays an important role during the motion matching procedure. In this thesis, we propose to learn features from motion data instead of designing features since hand-crafted features are not comprehensive enough to represent different kinds of motions. Motivated by the recent advancement of sparse representation which is commonly used to solve computer vision problems, we propose a temporal sparse representation (TSR) for human motion retrieval. Compared with existing methods that adopt sparse representation, our TSR encodes the temporal information within motions and thus generates a more compact and discriminative representation. In addition, we propose a spatial temporal pyramid matching (STPM) kernel based on TSR, which can be used for logical comparison between motions. Our STPM improves the effectiveness of motion retrieval in terms of accuracy and speed. To allow the user to retrieve desired motions in a natural and intuitive way, we develop a touch-less interactive human motion retrieval system. The system allows the user to specify the query motion by performing it directly with Kinect. Besides, the user interacts with the retrieval system using gestures so no controller is needed and the system delivers a natural user interface.

With the retrieved similar motions, we synthesize variations that can be used for intended applications. *Human motion variation synthesis* is important for crowd simulation and interactive applications to enhance the synthesis quality. Here, we propose a novel generative probabilistic model to synthesize variations of human motion with the retrieved similar motions. Our key idea is to model the conditional

distribution of each joint via a multivariate Gaussian Process model, namely Semi-parametric Latent Factor Model (SLFM). SLFM can effectively model the correlations between degrees of freedom (DOFs) of joints rather than dealing each DOF separately as implemented in existing methods. Detailed evaluation is performed to show the proposed approach can effectively synthesize variations of different types of motions. Motions generated by our method show a richer variations compared to those generated by existing methods.

Besides retrieving motions from pre-recorded motion capture database, *human posture reconstruction* with low cost device is an alternative way to obtain human motions. Recent research works show that devices that can estimate 3D postures from a single depth image (e.g. Kinect) have made interactive applications more appealing. In addition, it is rather costly to obtain the postures with mark-based motion capture technology. Hence, it is necessary to develop a robust method to reconstruct human posture using Kinect. Yet, it is still challenging to estimate pose accurately from a single depth camera due to the inherently noisy data derived from depth image and self-occluding action performed by the user. Here, we present a probabilistic framework to enhance the accuracy of the postures live captured by Kinect. We apply the Gaussian Process model as a prior to leverage position data obtained with Kinect and pose data from marker-based motion capture system. We also incorporate a temporal consistency term into the optimization framework to minimize the discrepancy between the current pose and the previous ones. Experimental results demonstrate that our system can achieve high quality postures even under severe self-occlusion situations, which is promising to be used for real-time posture

based applications.

Our proposed methods can free the user from generating realistic human motions and capturing new human movements with cheap device. With the proposed methods, the user can either retrieve motions from a large collection of motion capture database or synthesize similar motions based on the proposed variation synthesis approach. Moreover, the proposed posture reconstruction system allows the user to capture high quality human motions. Our methods are promising to be applied in computer games and animations to enhance the animation quality by introducing realistic human motions.

CITY UNIVERSITY OF HONG KONG

Qualifying Panel and Examination Panel

Surname:	ZHOU
First Name:	Liuyang
Degree:	Doctor of Philosophy
College/Department:	Department of Computer Science

The Qualifying Panel of the above student is composed of:

Supervisor(s)

Dr. LEUNG Wing Ho Howard	Department of Computer Science City University of Hong Kong

Qualifying Panel Member(s)

Dr. WONG Hau San	Department of Computer Science City University of Hong Kong
Prof. LI Qing	Department of Computer Science City University of Hong Kong

This thesis has been examined and approved by the following examiners:

Dr. LEUNG Wing Ho Howard	Department of Computer Science City University of Hong Kong
Dr. NGO Chong Wah	Department of Computer Science City University of Hong Kong
Prof. LI Qing	Department of Computer Science City University of Hong Kong
Prof. YUEN Pong Chi	Department of Computer Science Hong Kong Baptist University

Acknowledgement

I would like to express my deep gratitude to my supervisor, Dr. Howard Leung for his patient guidance, constant support and much encouragement. Dr. Leung led me into the research area of human motion analysis and synthesis. When I encountered difficulties in my research, he always gave me great ideas and valuable suggestions. His endless enthusiasm and deep insight into research influenced and will always influence me greatly.

I would also like to express my sincere gratitude to my Ph.D. qualifying panel members Prof. Qing Li and Dr. Raymond Wong for their valuable suggestions, which greatly contribute to my research studies in the past four years. I would also like to thank Dr. Huert Shum from Northumbria University and Dr. Zhiwu Lu from Renmin University of China for their suggestions that have inspired me a lot.

I thank my current and past lab mates Dr. Jeff Tang, Dr. Liqun Deng, Dr. Yang Yang, Dr. Jacky Chan, Miss. Lingling Yang, Mr. Zhiguang Liu, Mr. Billy Chiu and Mr. John Tam for broadening my views and sharing ideas with me. I am grateful for the support of my friends Mr. Liang Tao, Mr. Xudong Mao, Mr. Zheng Ma, Mr. Weichen Zhang, Mr. Ying Cao, Dr. Lifeng Shang, Mr. Yan Cai, Mr. Sijin Li, Mr.

Adeel Mumtaz, Mr. Chen Li and Mr. Bo Chen, for their help in my research and life in Hong Kong.

I wish to express my deepest gratitude to my parents and my sister. Their unconditional love and support have helped me to grow up all these years. I would also like to express my appreciation to my girlfriend Scarlet Li for her encouragement and patient accompany during my Ph.D. study. I am truly blessed with family and girlfriend, without whom I would never have completed my study. To them this thesis is dedicated.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

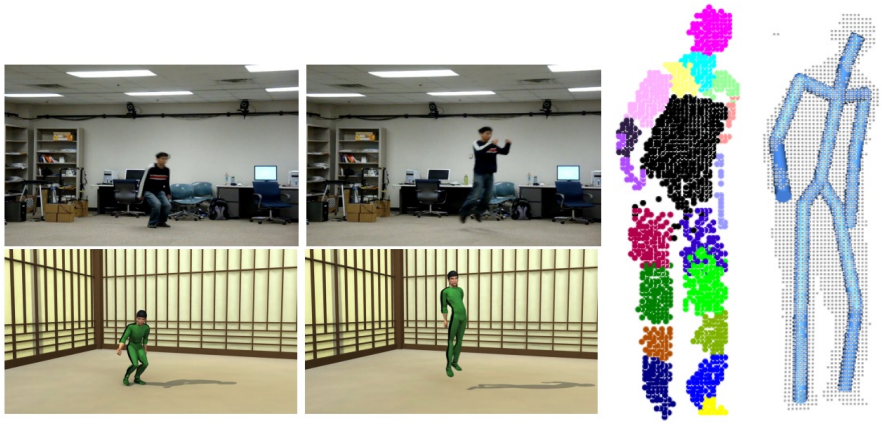
MOCAP	Motion Capture
PCA	Principal Component Analysis
SVD	Singular Value Decomposition
ICA	Independent Component Analysis
GP	Gaussian Process
TSR	Temporal Sparse Representation
STPM	Spatial Temporal Pyramid Matching
SLFM	Semiparametric Latent Factor Model
GPLVM	Gaussian Process Latent Variable Model
SOM	Self Organizing Map
DTW	Dynamic Time Warping
JRD	Joint Relative Distance

motions for intended applications.

With the development in motion capture technology, there are public motion capture database such as CMU05 [9], HDM05 [10] etc. To search for a specific motion from a large collection of motion capture data, an efficient retrieval mechanism is essential. Synthesis of human motion is another direction to reuse motion capture data. Here, we focus on human motion variation synthesis that can be used for crowd simulation. In this chapter, we introduce motion capture technology, followed by human motion retrieval, human motion variation synthesis and posture reconstruction.

1.1 3D Motion Capture

There are three methods for motion capture, namely vision based motion capture, optical motion capture and non-optical motion capture. Vision based motion capture consists of video-based motion capture and depth camera based motion capture. Video based motion capture is the technology to estimate pose sequences from either a monocular video or 3D videos. Such technology is preferable for surveillance applications. The basic idea is to subtract human silhouette from background and use different feature representation to estimate body joints. Figure 1.1(a) shows an example of video based motion capture, where the top two pictures represent the video sequence and the bottom pictures show the corresponding animations generated from the estimated motion data from the video. Recently, with the development of depth camera, RGB-D motion capture has emerged as a hot research topic. Depth camera provides more information of the human body such as the relative distance



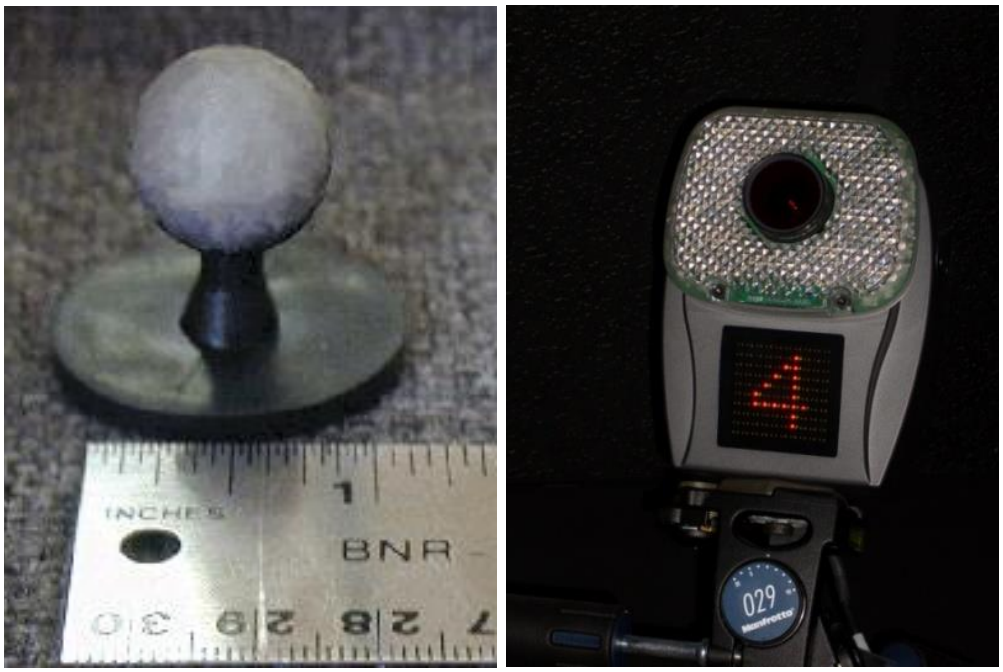
(a) (b)

Figure 1.1: Vision based human motion capture. a) Video based motion capture [1]; b) RGB-D based motion capture [2].

to the camera's location. Figure 1.1(b) shows an example of estimated poses from depth images. Although vision based motion capture technology is widely discussed and shows convincing results, the accuracy and robustness are not as satisfactory as optical motion capture system or non-optical motion capture system.

1.1.1 Motion Capture Devices

Optical motion capture system captures movements of the user by triangulating the 3D position captured by one or more cameras, and these cameras provide overlapping projections. The subject usually wears a tight suit with reflective markers on his body. For example, Figure 1.3(a) shows the subject wearing a black tight suit with markers on his body, and the markers represent the joints of the human body. These markers can either be active or passive. The active marker can radiate lights which can be detected by the system. Made of reflective material, the passive marker reflects lights radiated from infrared camera. Photos of the passive marker and infrared camera are



(a) (b)

Figure 1.2: Facilities of optical motion capture system. a) Reflective marker; b) Infrared camera.

shown in Figure 1.2.

Non-optical motion capture system uses sensors to record movements of human body. There are three types of sensor based motion capture system, which are mechanical, magnetic and inertial system as shown in Figure 1.5. In mechanical system, orientation sensors and potentiometers are put on the body of the performer to track his/her movements and are often considered as exoskeleton motion capture system. The user wears the device and performs motions to articulate the mechanical parts, and the system can measure the user's relative movements. There is no range limitation of the user's movement since the user carries the devices with himself, and it is also less costly compared to optical motion capture system. The device is quite heavy and hence not convenient for the user to perform fast and complicated motions. The

magnetic motion capture system has magnetic markers that are put on the subject and the system captures the motions of the performer in a magnetic field. The range of magnetic sensor is limited and it is sensitive to the noise of the environment such as the noise made by other electric devices. Inertial motion capture system uses gyroscopes to measure the rotational rates of the movements and these rotations are translated to be a skeleton in the system, together with biomechanical models. The inertial system is quite light and convenient for the user to carry, and the capturing volume is relatively large. However, the motions tend to be floating, where the performer looks like a marionette on a string. In addition, the lower accuracy of position will compound over time, which dramatically degrades the accuracy of the system. Considering all the above matters, we use an optical motion capture system in our laboratory, which is more stable and accurate to capture human motions.

1.1.2 Procedure of Motion Capture

The optical motion capture system used in our laboratory consists of 7 infrared cameras, which are installed at different locations on the wall so that together they cover the range of the capturing area, see Figure 1.4. The performer should do motions in a specified area, otherwise there will not be enough cameras to detect the markers and the system will fail to recover the 3D position of these markers.

In our system, there are 35 reflective markers on the tight suit the performer is wearing, each of which representing a joint of the human skeleton and some other body parts so that the system has sufficient data to articulate and recover the whole movement as shown in Figure 1.5(a). We can observe that these markers turned

(a) (b)

(c) (d)

Figure 1.3: Typical motion capture devices. a) Optical based device; b) Mechanical device; c) Magnetic device; d) Inertial device.

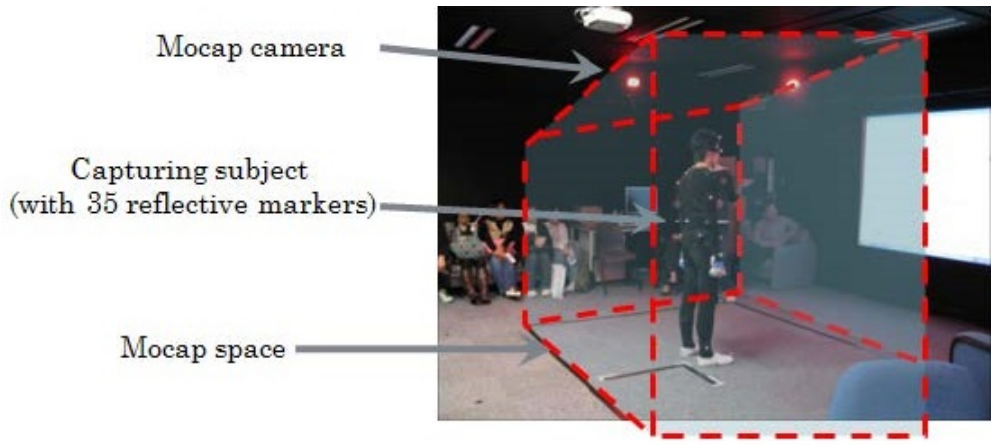
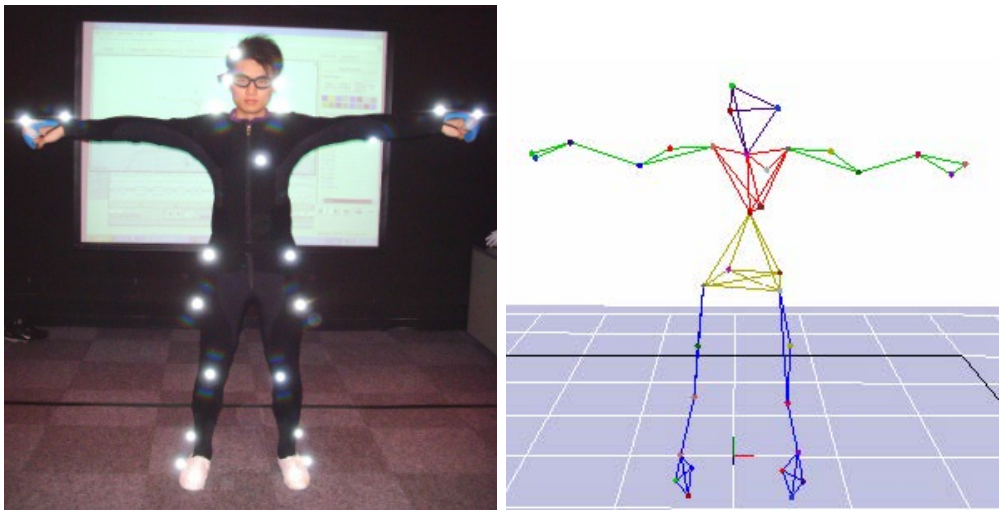


Figure 1.4: The motion capture facilities.

to be white as the markers reflect the flash light of the camera during photo taking. Figure 1.5(b) shows the visualization of the digital format of the motion data captured by our optical motion capture system. The dots of the skeleton correspond to the reflective markers and the lines between dots correspond to the bone segments.

Before actual motion capture, we need to calibrate the system to set up the range of capturing area and the coordinate system. The first step of calibration is to set up the coordinate system. We use a L-stick with markers on it to determine the coordinate system. Here the short stick corresponds to the x axis, long stick represents y axis, and the vertical direction corresponds to the z axis. The second step of calibration is to specify the range limit of the capturing volume. The user uses a T-stick with markers scanning the area and the system tracks the trajectories of the marker to determine the capturing area. Moreover, we manually specify which marker corresponds to the joint of a pre-defined skeleton, and the purpose of this step is to allow the system to recognize the markers.

With the above procedure, the optical motion capture system can estimate the



(a) (b)

Figure 1.5: The capturing subject and the corresponding captured skeleton. a) Optical markers adhered on the body; b) Captured 3D posture represented by a skeleton.

3D positions of the markers. On one hand, due to the occlusion problem, there will be artifacts of the obtained motion data, because the marker will not be tracked if there are not enough cameras for detection and the system will consider it as a missing marker. On the other hand, if two markers are too close during the movement, the system will mix up these two markers and induce errors. Thereby, we need to clean up the data to make sure the movement is smooth and there is no marker missing. One example of raw data is shown in Figure 1.6, which is of the right elbow joint. The curves at bottom represent the x, y, and z coordinates of the joint. It can be observed from these curves that they are not smooth and certain segments are split up because this marker was occluded and not recognized by the system during the movement. We need to manually correct and clean up the noise frame by frame. The task is rather time consuming as you have to check each frame and each joint manually, especially for the motions with frequent occlusions. Here, we use the software EvaRT to post-

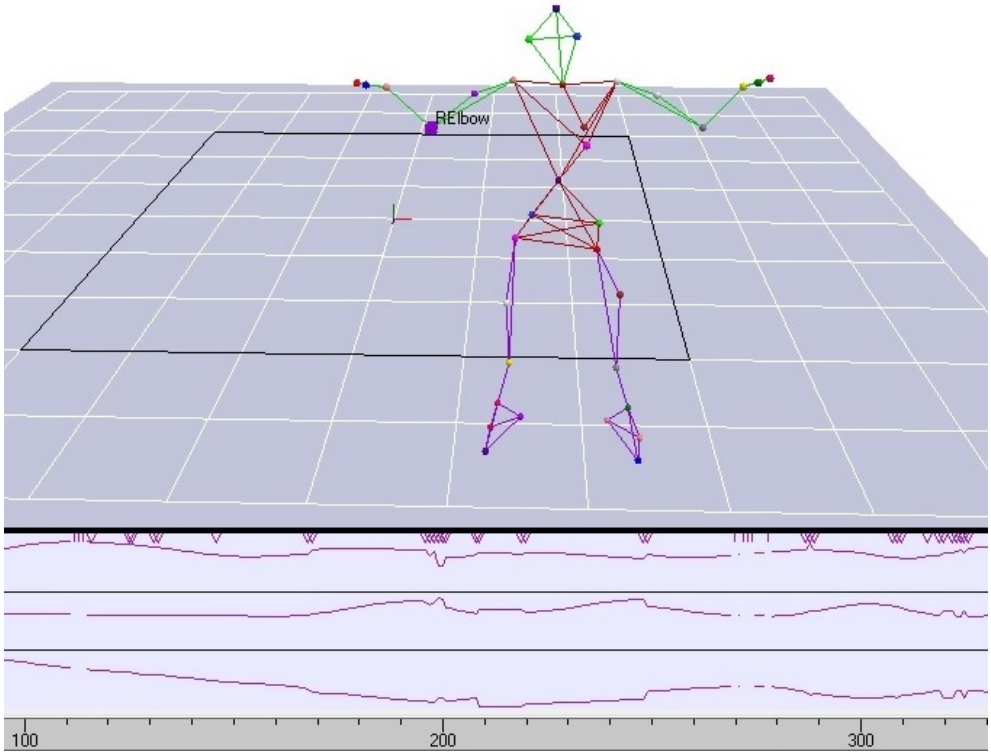


Figure 1.6: The raw motion capture data of the right elbow joint.

process the artifacts. The cleaned up data of the right elbow joint (i.e. Figure 1.6) is shown in Figure 1.7. Comparing Figure 1.7 with Figure 1.6, we can observe that the curves in Figure 1.7 are smoother and there is no missing segment.

1.1.3 Motion Data Formats

In this section, we give details to the motion data format used in our projects. The frame rate of the motion data depends on the motion capture device, which can be either 60 frames per second or 120 frames per second. There are two kinds of data used in this thesis. One is 60 frames per second, which is captured in CityU motion capture laboratory. The other one is 120 frame per second from public data set, such as CMU05 [9] and HDM05 [10]. In CityU motion capture laboratory, the

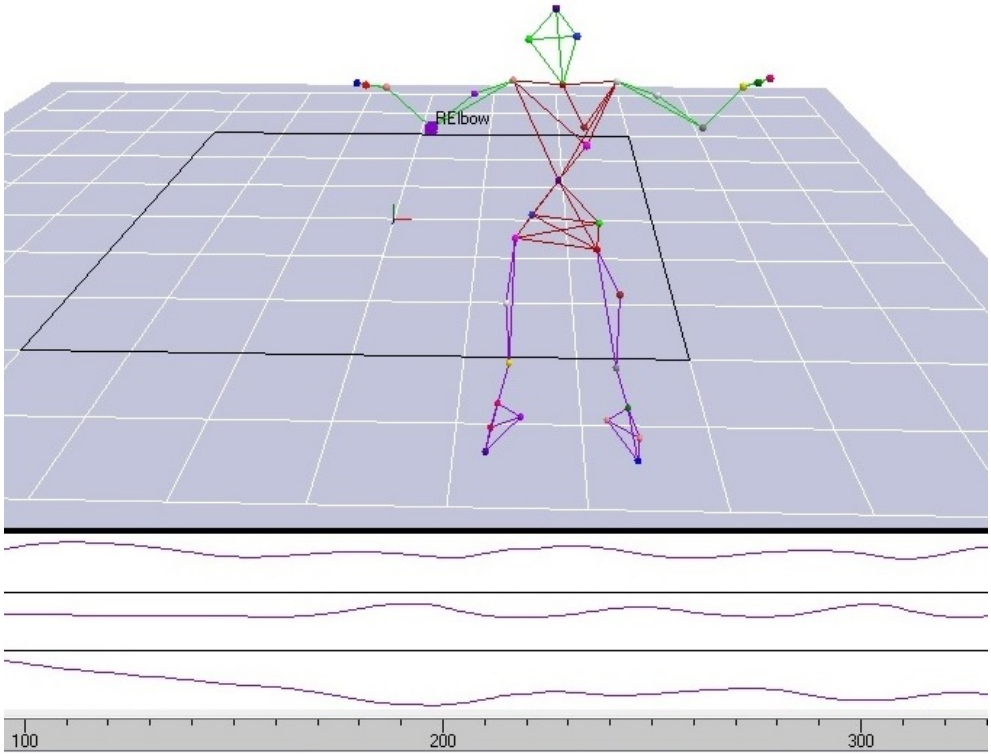
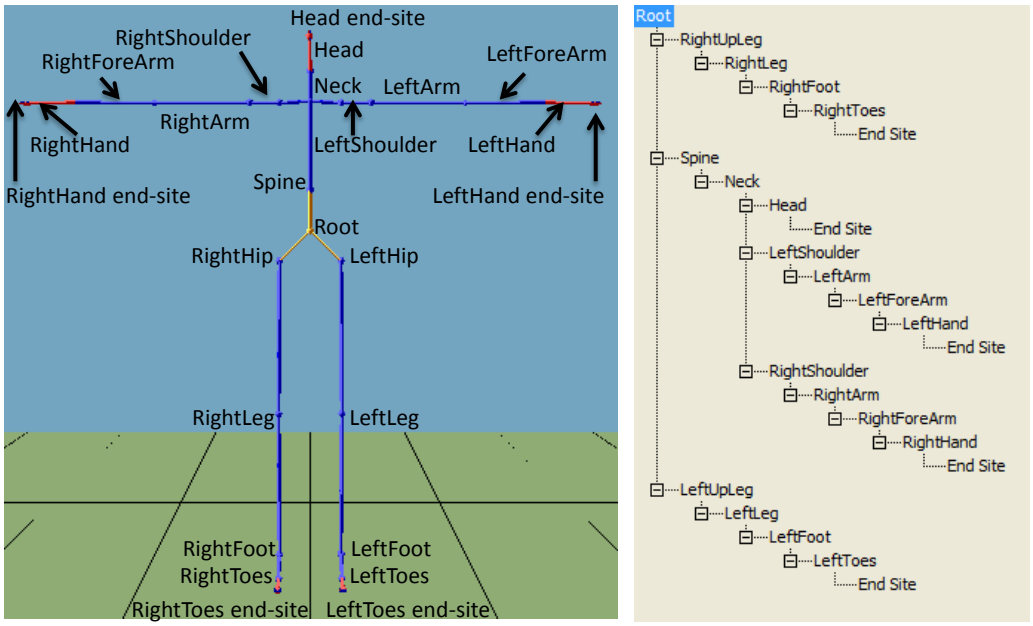


Figure 1.7: The cleaned up motion capture data of the right elbow joint in Figure 1.6.

optical motion capture system is developed by Motion Analysis Company. The output motion data format is called *MotionAnalysis TRC* format (.TRC). One example of TRC data is shown in Appendix A. The header of a TRC file describes the detailed configuration of the data, including data rate, number of frames, number of markers and units, etc. The data part corresponds to the x, y and z coordinates of each joint at each frame. For example, the dimension of data part at each row is 105 given that there are 35 markers.

However, TRC is the estimated 3D position of the markers, which more or less deviates from actual postures performed by the subject. The reason is that the markers are attached to the surface of the human body instead of the human body, e.g. skeleton. TRC format represents the actual position of the markers instead of

the real body joints. Hence, it will be more robust to estimate the joint position of the skeleton by mapping TRC to a pre-defined character model. The joint position can be calculated by the nearby joints with the Kinematic constraints such as bone length constraint, skeleton hierarchy constraint etc. Usually, people adopt *Biovision Hierarchy* (.BVH) and Acclaim Motion Capture (AMC) to store the joint rotation and skeleton information. Here, we use BVH format in this thesis as AMC needs another file to store the predefined skeleton information, namely Acclaim Skeleton File (ASF). We use the commercial software Autodesk MotionBuilder to convert TRC format into BVH format. One example of BVH data is shown in Appendix B. The BVH file contains ASCII text, the header of which specifies the start pose of one movement, and the information of the skeleton. The information of the skeleton consists of the bone length, which is indicated by the offset between two joints. Moreover, it defines the skeleton hierarchy between joints. Specifically, it provides parent-child relationship between joints. The data part contains the root position and poses information of other joints. The world position of the root joint provides the 3D world location of the skeleton. The position of other joints can be calculated by the transformation information specified by the offset and rotation related to its parent joints. In this thesis, we pre-define a skeleton with 20 joints and 5 end-sites as shown in Figure 1.8(a). However, such definition of skeleton hierarchy is different from the public motion capture such as CMU05 [9], HDM05 [10] etc. Hence, we need to carefully select and convert different skeleton hierarchies into the same skeleton structure. The chosen joints and the definition of skeleton depend on the application. For example, in Chapter 3, we use the skeleton based on the skeleton definition of



(a) (b)

Figure 1.8: a) The skeleton definition in CityU mocap system; b) The corresponding joint hierarchy structure of BVH format.

Kinect.

1.2 Motion Retrieval

Based on the description in Section 1.1.2, reusing motion capture data is a sound method as it is rather costly to capture new human motions. A number of research domains that reuse the pre-captured motion data have emerged, such as human motion synthesis, motion style translation and motion editing etc. These applications need an efficient retrieval mechanism to search for a specific motion sequence from a large motion repository, which has been proven to be quite challenging as human motion is high dimensional in both spatial and temporal domains. In addition, similar motions can be different in temporal length because of non-uniform speed differences,

which makes it more difficult to conduct similarity comparison between motions.

It should be noted that effective feature representation plays an important role in human motion retrieval. In literature, many efforts have been made to design new features to represent human motion. Müller et al. [11] introduced geometric features to describe the relations between body parts. Kapadia et al. [12] proposed motion keys to encode motions, where motion keys represent the structural, geometric and dynamic features of human motion. Tang et al. [13] used joint relative distance to describe pose sequences. However, these designed features are not comprehensive enough to describe different kinds of motions. Recently, sparse representation has emerged as an effective way to solve many problems in computer vision. In this thesis, we learn features from data via sparse coding instead of feature designing. It is suitable to use sparse representation for human motion data, as both the spatial and temporal domains of human motion are highly correlated, which results in their sparseness [14]. Sparse coding can achieve fewer reconstruction errors compared with traditional vector quantization methods. Besides, sparse representation can capture the salient properties of data [15]. Sparse representation consists of two steps, namely dictionary learning and sparse coding [16]. The original data can be represented as linear combinations of the atoms in the learned dictionary, where the coefficients are called sparse representation through the step of sparse coding. Sparse representation is combined with bag of features for image and video classification. However, it either destroys the spatial information for images or the temporal information of videos. Spatial pyramid matching [15] is thus used to solve the spatial order problem by dividing an image into segments in different scales.

Here, we also adopt sparse coding to obtain the sparse representation of human motion data. Unlike the traditional sparse representation that ignores the temporal information of human motion, we propose a novel temporal sparse representation (T-SR) to capture the temporal information within one motion. Our TSR considers the temporal relationships between frames of sparse coefficients. Similar idea has been adopted in [17] to capture the spatial information within images by considering the spatial dependency between visual keywords. In our case, we consider the temporal dependency within one motion by the outer product of two frames from sparse representation. It implicitly takes into account the relationships between one atom in one frame and all the other atoms in another frame.

Our TSR is a 3-D array of size $M \times M \times N$, each slice being a square matrix ($M \times M$) and representing the relations between atoms while N being the length of TSR. However, such local temporal information is still not enough to encode motions and it is difficult to compare between TSRs as they are 3-D arrays. We further divide our TSR into spatio-temporal cuboids in both spatial and temporal domains at different scales. We also derive our spatial temporal pyramid matching kernel (STPM) by comparing TSR in different temporal and spatial scales, which greatly improves the accuracy and efficiency for motion comparison.

Exemplar based human motion retrieval methods are not applicable when the user does not have the query motion on hand. Numaguchi et al. [18] proposed a puppet interface for human motion retrieval, in which the puppet has 17 degrees of freedom. Feng et al. [19] proposed a keyframe-based human motion retrieval system which used a wooden doll to capture the query motion as input. These devices are

not flexible to express the dynamics of the motion. Besides, it is not convenient to manipulate these devices as the user has to hold them to express the query motion. Meanwhile, sketch based method is another way to retrieve motions. For instance, define the motion by sketching several strokes over a drawn character [20] [21]. Yet sketch based methods are not suitable for novices as the user has to know how the system represents motions with drawings. In addition, sketching styles vary among different users, which will certainly affect the result of motion retrieval.

Human motion tracking hardware has made gesture control applications and video games more interactive and addictive [22] [23] [24]. In this thesis, we propose a touch-less human motion retrieval system based on Microsoft Kinect. On one hand, the system provides sample motions in the database for the user to choose as a query. On the other hand, the system allows the user to provide the query motion by performing the query motion. It is more natural and intuitive to capture the query motion with Kinect since the user knows what kind of motion he/she wants. In addition, the proposed controller-free interface allows the user to control the system with different gesture commands, and is therefore more user-friendly than traditional mouse-keyboard input devices.

1.3 Motion Synthesis

Human motion synthesis is essential to generate human motions for different applications. There are a lot of researches in human motion synthesis area that depend on the intended applications, which consist of partial human motion synthesis [25],

responsive human motion synthesis [26], human-object motion synthesis [27], interaction synthesis [28] and human motion variation synthesis [29]. In this thesis, we focus on human motion variation synthesis. There is a high demand for variation synthesis in character animation and game domains, since the quality of animation could be affected by the recurrent motion clones [30]. We observe that real human subjects tend to perform the same action differently each time. Although motion capture technology can be utilized to record a person's movements, it is not practical to apply this technique to capture different variations of the same motion, as it costs time and labor. Therefore, it is important to develop an effective method to automatically generate variations that are based on a small set of example motions.

Human motion variation synthesis can be viewed as a special case of human motion synthesis [31]. Style transferring is the way to synthesize a new style of motions by transferring style from one motion to another [32]. However, most style transferring techniques cannot be used to synthesize variations within the same style. Motion interpolation extracts style parameters to synthesize motions using interpolation [29], whereas style parameters are not intuitive and usually difficult to be extracted. Another way to generate new variations is motion editing, which is editing existing motions [33]. However, this technique usually requires a manual tuning procedure, which is not suitable for novice users. In other words, human motion variation synthesis can be considered as a one-to-many mapping procedure, while existing methods mainly focus on one-to-one mapping.

In this thesis, we propose a novel generative probabilistic method to generate a large number of new variants based on a small set of example motions. We divide the

kinematic skeleton into multiple partitions based on the human skeleton hierarchy, which not only reduces the complexity of human motion but also helps to model the relations between joints. We predefine the influence between joints within the same body partition based on the hierarchy of the skeleton structure. Such influence is translated into the conditional dependency relations between joints. The conditional probability distribution for each joint is calculated by Semiparametric Latent Factor Model (SLFM) [34], which differs from standard Gaussian Process (GP) since SLFM can capture the dependency between multiple outputs. SLFM is an extension of the generally used univariate GP for regression problems involving multiple response variables. The basic idea of this model is to use a set of basic GPs and then linearly mix them to capture dependency that may exist among the output variables. In our situation, this model can effectively capture the relations between different DOFs within one joint. New variants can therefore be synthesized by sampling from the predicted distribution. Besides, compared to other non-parametric regression methods such as kernel regression, GP based model can robustly learn from small training sets and the parameters of kernel function can be optimized without relying on experimental cross-validation, such as the kernel width for kernel regression. Most importantly, our synthesized motions show more variations compared with existing methods, which is because we directly use the skeleton configuration feature as target instead of the generally used frame difference [8].

1.4 Human Posture Reconstruction

Human posture recognition is the core part of interactive applications. Chan et al. [35] proposed a dance training system based on motion capture technology, where the user's movements are captured by an optical motion capture system. The dance training system allows the user to learn the dancing movements interactively. While these applications can evaluate user performance by accurately capturing user's motions, they are not convenient since the user has to wear a tight suit with reflective markers on it. Wii-mote device (see Figure 1.9(a)) services as an alternative to capture movements and it is used for interactive applications. For example, Schlomer et al. [36] used a Wii controller for gesture recognition. Wingrave et al. discussed the capability of Wii mote as a 3D user interface [3]. Unlike the dance training system proposed by Chan et al. [35], the user does not need to wear a suit, he/she needs to hold the device when performing the motions. Therefore, it is essential to develop an effective posture reconstruction method with controller-free device for interactive applications.

Recent advance in motion tracking devices that are based on depth camera such as Microsoft Kinect (see Figure 1.9(b)) have enabled efficient human-computer interaction using body movement, and enhance interactive systems such as console games. Kinect is a controller-free hardware that infers 3D positions of human body joints from a single depth image with the help of motion recognition technology [37]. It is convenient and intuitive to use Kinect for gesture based applications such as turning on/off light, switching slides etc. Kinect can be used to track the user and determine

(a) (b)

Figure 1.9: Simple posture capture device. a) Wii-mote [3]; b) Kinect [4].

the user's 3D joint positions in a robust manner, which is convenient for posture control. Yet the captured data suffers from poor precision due to self-occlusions and insufficient information provided by Kinect sensor, since Kinect can only detect the frontal direction related to the location of Kinect. As a result, the Kinect based interactive applications usually require the user to face the device as straight as possible so that the system can track the user's movements. In addition, the motions performed by the user should not contain much self-occlusion postures. The occlusion problem and incompleteness of the tracked joints remains challenging despite the relevant research proposed in the past years. Generating poses from low dimensional signals is one way for posture reconstruction [38] [39], which assumes low dimensional signals are stable. Hence, it cannot be applied to posture reconstruction of Kinect as the tracked joints from Kinect are not stable. Another way for reconstructing postures is to retrieve similar poses from a pre-defined motion capture database and use these retrieved postures for recovery. Nevertheless, it needs a large pre-defined database and the results will degenerate greatly if there are no similar postures in the database.

In this thesis, we propose a probabilistic model to reconstruct postures captured from Kinect. We adopt Gaussian Process (GP) model as prior distribution to estimate the most likely correct posture given one posture data from Kinect, which is called spatial prior. We record the postures with both Kinect and an optical motion capture system. We used the joint position information from Kinect, and it is in the same data space as marker-based motion capture data. Normal RGB cameras can be used for pose estimation, whereas it is time consuming for pose estimation and not suitable for interactive applications. In addition, calibration between cameras is another challenging problem. The relationship between pairwise data from Kinect and the motion capture system is extracted for modeling with GP. Unlike other systems that use a large marker-based motion database, GP based model can be robustly trained from small training sets. Moreover, the parameters of the kernel function can be optimized without relying on experimental cross validation. Reconstructing each posture independently cannot ensure the temporal smoothness of the pose sequence. To tackle this problem, we introduce a temporal consistency term to constrain the velocity variations between successive frames. To make sure the reconstructed pose matches the observed Kinect input data, we embed the reliability of each joint into the optimization framework. Specifically, we keep the joint value as original as possible if the tracked joint is reliable (correct). We very the effectiveness of our approach by reconstructing a number of motion containing self-occlusions.

1.5 Outline of the Thesis

The organization of this thesis is as follows. In Chapter 2, we provide related work in human motion retrieval, human motion variation synthesis and human posture reconstruction. In Chapter 3, we introduce temporal sparse representation for human motion retrieval and a touch-less human motion retrieval system. In Chapter 4, we present multivariate prediction framework for human motion variation synthesis. In Chapter 5, we present a probabilistic framework for human posture reconstruction. We conclude this thesis and discuss about future work in Chapter 6.

Chapter 2

Related Work

It is labor intensive to use motion capture data since it consists of two time-consuming steps, which are capturing motions and post processing artifacts. Therefore, it is essential to develop methods to either reuse pre-captured motion data or synthesize new human motions for different applications. However, it is a challenging problem as human motion is high dimensional multivariate time series data. In the past years, there are many methods have been proposed to solve this problem. In this chapter, we review the related work on single human motion retrieval (Section 2.1), human motion variation synthesis(Section 2.2) and human posture reconstruction(Section 2.3).

2.1 Human Motion Retrieval

Human motion is highly coordinated time series data and it is not efficient to match between two motions directly. A number of methods have been proposed for human motion retrieval. One category is to construct index structure or transform motion

data into low dimensional space to speed up retrieval. Another way is to use new feature representation to extract the salient property of one motion so that the motion matching procedure will be more efficient. The query specification is also important so that the user can retrieve motions in an intuitive way. In the following of section, we will elaborate the details of each category.

2.1.1 Transformation based Motion Retrieval

Transforming the original motion data into low dimensional space and index construction of the motion database to speed up the retrieval procedure is widely studied recently. Principle component analysis (PCA) and singular value decomposition (SVD) are widely used as a dimension reduction method for retrieval purpose. Pradhan et al. [40] constructed an index structure based on the skeleton hierarchy, and then Singular Value Decomposition (SVD) was used to map the human motion data into a low dimensional feature space at each level of the index. The matching between motions is now transformed into low dimensional signal matching for human motion retrieval. Jin et al. [41] used Gaussian Mixture Model (GMM) to quantize human motion based on the center of each cluster, and K-Nearest Neighbor (KNN) was adopted to find the best single matching motion. Forbes et al. [42] presented a weighted-PCA pose representation and approximate nearest neighbor (ANN) search was used for retrieval. Wang et al. [43] performed the matching on individual body parts as well as on the whole body for pruning irrelevant motion, in which each motion was represented by eigen vectors. However, such dimension reduction methods usually end up losing the nonlinear information of the human motion data as well as

a lot of detailed information of the movement, which makes the retrieval procedure rather coarse and inaccurate. Recently, Sun et el. [6] proposed a low-rank decomposition approach to convert the motion sequence volumes into lower dimensional representations without losing the nonlinear property of the motions.

Index construction of the database is another way to speed up the retrieval. Liu et al. [44] partitioned the motion library and constructed a hierarchical index tree, which served as a classifier to find the candidate sequences to the query example. Tanuwijaya et al. [45] transformed the motion data into textual search problem, where term frequency and inverse document frequency indexing were used to speed up the retrieval. Huang et al. [46] decomposed motion files into kinetic intervals, where the kinetic interval feature was defined as parameters of parametric arc equations computed by fitting joints trajectories. Multilayer index tree is used to accelerate the searching process based on the kinetic interval features. Wu et al. [47] used self-organizing map (SOM) for index and each motion was represented by the nodes of the map to get the motion strings. The motion matching problem was transformed into string matching problem, which was solved by the smith-waterman algorithm. Chiu et al. [48] proposed an index map structure based on the posture distribution of raw data. The similarity between the query example and candidate clips was computed through dynamic time warping (DTW). Deng et al. [5] break human motion into part-based and hierarchical motion representation, where KD-tree based motion pattern library is used for storing the details of the extracted motion patterns. Building upon this representation, Knuth-Morris-Pratt string matching algorithm was used for runtime query processing. Although index based methods can be used to speed up

motion retrieval, they usually neglect the temporal attribute of the motion sequence, whereas temporal information is one important property of human motion data.

2.1.2 Feature based Motion Retrieval

As presented in [49] [11], logical similar motions are not necessarily numerically similar. Various features have been proposed to represent human motion in a semantically way for logically human motion retrieval. Müller et al. [11] introduced geometric features to describe the relations between body parts. For instance, one geometric feature represents whether or not the left foot lands in front of the plan spanned by the right foot. Recently, Kapadia et al. [12] proposed motion keys to encode motions, which included not only geometric features but also dynamic features. Multiple combinations of motion keys can be used to facilitate retrieval of complex motions. However, the user has to specify suitable geometric features or motion keys to better describe the motions, which is not suitable for novices since the user need to know what kind of keys or geometric features are used to represent motions beforehand. Besides, such user-determined features are limited and incapable of expressing a large variety of motions. Tang et al. [13] proposed joint relative distance (JRD) as a new feature representation to emulate the perception of motion similarity. Tang and Leung [50] proposed an adaptive feature selection method built upon JRD, which made the subset feature selection can be according to the properties of the specific query. Chen et al. [51] extend JRD to be geometric pose descriptor by utilizing features on geometric relations among body parts. Clustering methods such as Gaussian Mixture Model (GMM) can be used to quantize the human motion data to extract

semantic features, where each frame is represented as the center of each motion class. Then histogram representation can be extracted for the similarity comparison between motions [52] [53]. However, hand crafted features are not sophisticated enough to represent different kinds of human motions.

Learning features from data other than designing features is another way to extract human motion representation such as sparse representation. Sparse representation has been proven to be a compact and discriminative feature representation. It has been successfully applied in various fields such as image denoising [16], face recognition [54], video based human action recognition [55] etc. The basic idea of sparse representation is to represent the signal as a linear combination of a set of atoms (called dictionary), where the coefficients are constrained to be as sparse as possible. As a result, the sparsity allows the representation to be specialized and also to capture the salient part of the data [15]. In our motion retrieval system, sparse coding is adopted to convert the original motion data into its sparse representation. The joints of human body are highly correlated in spatial domain and the frame rate is high in temporal domain. Hence, it is reasonable to assume that human motion data is sparse. Zhu et al. [7] introduced quaternion space sparse decomposition for human motion compression and retrieval. The similarity between motions is obtained through comparison between the dictionaries decomposed from each motion respectively. In our approach, we obtained the sparse representation in the Euclidean space. The Euclidean space is more intuitive to measure the errors, since we can minimize the square errors of the sparse representation directly. In addition, it avoids the periodicity of angles, which potentially corrupts the sparse

representation [56]. Unlike the traditional sparse representation, we proposed temporal sparse representation that encodes the temporal information by considering the relationship between frames of sparse representation, which is more discriminative as a human motion representation.

2.1.3 Query Specification Interface

One simple way to retrieve motions is through text matching, where each motion is annotated with key words [9]. However, this approach is insufficient due to the complexity of motion data. Recently, external props were used for query motion specification. Feng et al. [19] presented a key frame based human motion retrieval system, in which a wooden doll was used as the input device. The user inputs a key frame by posing a doll with painted joints in front of a monocular camera. It is suitable when there is no query example motion. However, the user needs to input each key frame with an artist's doll, which is not flexible for the query specification. Numaguchi et al. [18] presented a puppet interface for human motion retrieval, where the puppet has 17 degrees of freedom. The similarity between puppet and human motion was computed by the reconstruction errors of projecting the puppet motion and human motion into each other's latent space. Although it is more convenient to capture a motion as the query compared with commercial motion capture system such as optical motion capture system, it cannot capture the dynamics (i.e. translation) of the motion. Furthermore, it is not convenient to manipulate these props.

Another intuitive direction is sketch based motion retrieval, where motions are specified through sketches. Choi et al. [20] proposed to represent human motions as

2D stick figure sequence, where the tick figure represent the poses and the curves corresponding to the dynamic property of the movement. Chao et al. [21] presented sketch based motion retrieval system, where the system define the required motion by sketching motion strokes over a drawn character. The user can draw lines of each certain joint to represent the movement trajectory. However, not all 3D motion curves can be easily described from a 2D drawing such as hip hop dancing.

Tang et al. [57] adopted a similar idea to project 3D postures into 2D planes, and embedded limb direction into the feature representation. The user can retrieve desired motions by drawing 2D figures. Although curves were introduced to represent moving joints, they cannot describe the continuity of human motion sequence. Besides, the drawing styles vary between different users, which may thereby affect the retrieval result. Kapadia et al. [12] proposed a motion retrieval system that allows the user to use Kinect for query specification. However, the inputs of the user's gestures are motion keys instead of the motion data directly, which is not intuitive and convenient for novices. In this thesis, we propose to use Kinect as a simple but intuitive device to capture the query motion of the user. Kinect is widely used in interactive applications because of its freehand control property [22] [24] [23]. Although Kinect cannot capture motions as accurate as motion capture system due to the occlusion problem, we can still use it to capture one motion to be input as a query motion. Moreover, our controller-free interface uses different control commands, which allows the user to control the system with touch less interactions.

2.2 Human Motion Variation Synthesis

Human motion synthesis is the main research direction in the motion capture area. Here, we are particular interested in the work of human motion variation synthesis. Human motion variation can be defined as the differences within the same type of motions. For instance, when a group of people perform a punch motion, how they strike their fists and the strength of their punches can be distinctively different from each other. Some previous methods were developed for motion variation generation by adding noise to the existing ones [58]. However, variation is not merely noise or error, but rather a functional component of data itself [59]. A number of methods for generating new motions from example motions have been developed during the past years for different applications. These methods can be roughly categorized as interpolation based methods, linear statistical methods and nonlinear probabilistic methods. In the following of this section, we will elaborate the related work in each category.

2.2.1 Interpolation based Methods

A number of methods that employ interpolation related techniques to generate new motions from existing example motions have been developed. Rose et al. [60] characterized human motion by emotional expressiveness and control motions such as turning or going uphill or downhill. They defined these parameterized motions verbs and the parameters that used to control them adverbs. Radial basis functions and low order polynomials were used to create the interpolation space based on the verb graph

to generate new motions. The proposed method is general and scales well with the dimension of the adverbs. However, the system only accepts variations through the user's input. Safonova et al. [61] proposed to synthesize physically realistic variations provided by interpolation between two time-scaled paths through a motion graph. The purpose of the graph construction was to support interpolation and pruned for efficient retrieval. The system allows the user to specify motion by sketching the path of the character through the environment. However, the system require the poses for interpolation to be the same contacts.

Ma et al. [29] introduced latent variation parameters to parameterize the variation of each motion. The relationships between user-defined style parameters and latent variation parameters are learned by Bayesian Network. As a result, new human motions can be synthesized by the interpolation between these style parameters. However, such interpolation based methods have to extract style or user specified parameters, which may not be easily defined systemically, such as dancing motions. Unlike the above methods, our generative model can directly learn from the training data without extracting any interpolation parameters.

2.2.2 Linear Statistical Methods

Linear statistical methods have been widely adopted for motion analysis and synthesis. For example, Urtasun et al. [62] decomposed human motion into principle motions through principle component analysis (PCA). Motions are parameterized by different parameters. For example, walking and running motion were parameterized by speed, and jumping motion was parameterized in terms of jump length. New mo-

tion variants can be extrapolated by a set of best approximated principle weights from a set of similar motions with different parameter values. Min et al. [63] proposed to extract the timing and posture information separately for a bunch of similar motions. PCA was adopted to represent the timing and postures with principle components, and then Gaussian Mixture Model was used to model the distribution of these principle components. New motion variations can be generated by sampling from the obtained distribution. However, their purpose was for interactive applications and cannot generate a lot of variants.

Li et al. [64] proposed motion texture to synthesize statistically similar motion to the original human motion capture data. Motion texture was represented by a set of motion textons and their distribution. Linear dynamic systems were used to capture the local dynamics of motion textons. Although the proposed method can generate realistic and dynamic motion, the synthesized motion may lack the global variations when the training data is limited. Kim and Neff [65] used Independent Component Analysis (ICA) to divide the stylistic example locomotion into sub motion components. New stylistically different locomotion can be synthesized through the composition of the sub-motion components. Although linear statistical method is intuitive and easy to implement, human motions are high dimensional data that may behave in a non-linear manner and it is difficult to extract style parameters with a linear perspective.

2.2.3 Nonlinear Probabilistic Methods

Probabilistic based methods have been used to create new motions for different applications. Ikemoto et al. [66] proposed a motion editing system to generate new motions based on Gaussian Process (GP) model. The relationship between original motion and the one after editing is extract as the input and output of GP model. To enhance the motion quality, both kinematic and dynamic GP model is adopted to constrain the solution space. Although the proposed system can make the motion editing procedure easier and intuitive, it is not applicable to novices as it needs to know strong knowledge of human motion for editing. Grochow et al. [67] proposed style-based inverse kinematics system. Style based inverse kinematics was developed upon Gaussian Process Latent Variable Model (GP-LVM) [68]. The system can produce the mostly likely pose satisfying a set of user defined constraints. It can generate human motion variations when constrains are set to be statistically similar. However, the system did not model the dynamics and take into account the constraints that produced the smooth human motion. Wang et al. [69] augmented GP-LVM with a dynamic prior to get smooth trajectories in latent space, and a map from the latent space to the pose space, which is called Gaussian Process Dynamic Models (GPDM). Despite the use of small data sets, the GPDM learned an effective representation of the nonlinear dynamics of human motions in these spaces. These GP-LVM based models are powerful methods for human motion modeling, yet they are not suitable for our application as there is no guarantee that the latent space points are densely connected to synthesize human motion variations. In addition, GP-LVM is construct-

ed based on the independent assumptions between different observations dimensions. So the correlations between DOFs of joints cannot be modeled.

Probabilistic methods are one direction for motion modeling, but it seldom considers physical information of motions. Wei et al. [70] used Gaussian Process (GP) to model the nonlinear probabilistic force field function from prerecorded motion capture data and combined it with physical constraints. The proposed system can generate physically valid human motions and react to external forces or changes in the physical quantities of character body and the environment. Style learning and transferring can also be viewed as one way to generate variations. Many researchers have developed effective methods for style-based motion generation, including style machines based on Hidden Markov Models [71]. Wang et al. [72] proposed a multifactor GP model for style-content separation. Hsu et al. [32] transferred the input motion to a new style while preserving the original content. Style transferring is very similar to our purpose of generating new variations. The work of Hsu et al. [32] is about one-to-one mapping, and their purpose is to translate one style of motion to another style and it cannot generate many variations within one style of motion. Yet our purpose focuses more on generating variations within the same style of motion, which can be considered as a one-to-many problem. Specifically, given one style of motion, our purpose is to synthesize a bunch of variants that are similar to each other within the same style.

Among all the related research, Lau's work [8] is the most relevant to ours. They introduced Dynamic Bayesian Network (DBN) to model spatiotemporal variation of human motion. Conditional dependency is learned by a nonparametric kernel

regression technique from training data. Yet their method was highly affected by the tuning of kernel width and the selection of neighboring instances. Ma et al. [29] also used Bayesian Network (BN) to model the relations between user defined style parameters and extracted variation parameters. However, the structure of their BNs has to be manually adapted for different motions, which reduces the generalization ability of this method. In this thesis, the skeleton representation is divided into multiple partitions to obtain the dependency between joints. This partition based structure is more general as it does not need to consider the specific type about the motion.

2.3 Human Posture Reconstruction

With the advancement in real-time depth camera such as Kinect sensor, human motion recognition and pose estimation are widely discussed in recent years. Kinect sensor consists of an infrared sensor and combined with a monochrome COMOS sensor, which records video data and depth data. Kinect is based on motion recognition technology proposed by Shotton et al. [37] , where they use per-pixel classification method to quickly predict 3D joint positions from a single depth image. Kinect has fused a wide variety of research areas, such as human-machine interaction [73], natural user interfaces [74], and 3D reconstruction [75] etc. A recent review on human activity analysis with Kinect can be found [76]. Bailey and Bodenheimer [77] investigated the perceived differences in the quality of animation generated using motion capture data and a Kinect sensor, which showed the data recorded from Kinect is clearly with

low quality compared with motion capture data by a Vicon motion capture system. As a result, it is essential to develop an effective posture reconstruction method. In this section, we briefly review the related works about posture reconstruction.

2.3.1 Tracking based Posture Reconstruction

Human motion tracking can be considered as one way for posture reconstruction. Tracking based approaches usually require registering a 3D articulated model with depth information. Wei et al. [2] formulated the registration problem into a Maximum A Posteriori (MAP) framework to register a 3D articulated human body model with monocular depth via linear system solvers. They integrate depth data, silhouette information, full body geometry and temporal pose prior into a unified framework. To tackle with the problem of manually initialization and recovery from fails, they combined 3D pose tracking with 3D pose detection. Although the proposed algorithm is parallel and can be implemented on GPU to accelerate the speed, it requires the calibration procedure to make the system can be used for different skeleton sizes. On the contrary, our system is invariant to the location of the user and the skeleton size of the user.

Yasin et al. [78] proposed to use motion capture database as the prior knowledge for full body reconstruction from 2D video data. Two dimensional features were extracted from both the input video sequence and motion capture database. With the obtained features, postures were reconstructed in a data driven optimization framework, in which similar motion capture postures were retrieved through nearest neighbor searching. However, the accuracy is not robust as the projecting 3D motion

into 2D features will induce ambiguous problem of postures. Oikonomidis et al. [79] proposed a model-based 3D tracking method for hand articulations using Kinect, where the system minimizes the difference between a pre-defined hand model and the actual hand observations. Although the system can obtain robust tracking results, it lacks the ability of generalization since the whole framework relies on a predefined hand model.

To enhance the physical validation of reconstructed motions, dynamical models are adopted to constrain the solution space. For example, Taylor et al. [80] introduced a probabilistic latent variable model for human pose tracking, namely Implicit Mixture of Conditional Restricted Boltzmann Machines. Vondrak et al. [81] proposed to use a simulation-based dynamical motion prior for human motion tracking. These dynamical model based approaches are time consuming and are not applicable for Kinect based interactive applications.

2.3.2 Posture Reconstruction from Low Dimensional Signals

The full body postures can be represented by a set of low dimensional signals [38]. Some research work has been proposed to reconstruct a posture with a small set of signals. Kim et al. [82] reconstructed human motion from sparse 3D motion sensors on a performer using kernel CCA-based regression. Given the input data from 3D motion sensors, they retrieve similar poses from the motion capture database and an online local model was proposed to transform the input low dimensional signal into the pose space. Chai et al. [38] employs a small set of retro-reflective markers to create a performance control system, where the set of markers are used to find the most

matched motion examples for reconstruction. In their system, the low dimensional control signals from the user's performance were supplemented by a database of pre-recorded human motion. At run time, the system automatically learned some local models from the retrieved motion capture examples that were closed to the marker locations captured by the camera. Their system is low cost since it only needs video cameras and small set of markers, which is practical for home use. However, the user cannot move freely in the space as most of the markers needed to be seen by at least one camera.

Liu et al. [39] used a small number of motion sensors to control a full-body human character. They construct online local dynamic models from prerecorded motion capture database and use them to construct full-body human motion in a maximum a posteriori framework, in which the system tried to find the most like poses from database for reconstruction. Helten et al. [83] adaptively fused inertial and depth information in a hybrid framework for pose estimation. Although these methods can be used to reconstruct postures from low dimensional signals, there is one assumption that these low dimensional signals are reliable and stable. It is not applicable to Kinect data as poses from Kinect are not stable and consistent.

2.3.3 Data-Driven Posture Reconstruction

Kinect is a RGB-D sensor, which provides more information than monocular camera. Sigalas et al. [84] proposed to estimate 3D torso poses from RGB-D images based on a data-driven model. Their system showed good estimation for torso poses, however, their approach is not suitable for full body pose reconstruction nor handle the oc-

clusion problem. Shum et al. [85] proposed a unified framework to control physically simulated characters with live captured motion from Kinect by searching similar poses from marker-based motion database. They constructed posture space based on the retrieved similar postures. The postures were reconstructed with the posture space in an optimization framework. Baak et al. [86] introduced a data driven approach for full body reconstruction from a depth camera. They proposed a variant of Dijkstra's algorithm to extract the posture features from depth information and a novel late-fusion scheme based on computable sparse Hausdorff distance, which is used for local and global pose estimation. Although their approach can obtain good results, it requires the query procedure in database that increase the complexity of the system.

Shen et al. [87] introduced an exemplar-based method to correct the poses from Kinect using marker-based motion data. Data-driven methods need to construct a large motion capture database a prior knowledge. In this thesis, we use Gaussian Process (GP) to model the prior distribution of poses from Kinect with marker-based motion data. Our approach delivers the power and effectiveness even with small training data as GP based model can robustly learn from small training sets. Moreover, our method requires no manual intervention such as marker labeling in model based works.

Chapter 3

Human Motion Retrieval

To search for a particular motion from a large collection of motion capture data, an efficient retrieval mechanism is essential. This has been proven to be challenging as human motion is high dimensional in both spatial and temporal domains. Besides, semantically similar motions are not necessarily numerically similar because of the speed variations. In this chapter, we propose a novel temporal sparse representation (TSR) for human motion retrieval. Compared with existing methods that adopt sparse representation, the proposed TSR encodes the temporal information within motions and thus generates a more compact and discriminative representation. In addition, we propose a spatial temporal pyramid matching (STPM) kernel based on TSR, which can be used for logical comparison between motions. Moreover, it improves the effectiveness of motion retrieval in terms of accuracy. Through our experimental evaluations, we demonstrate that the proposed human motion retrieval system has better performance and allows the user to retrieve desired motions from motion capture database. Finally, we implemented a touch-less human motion re-

trieval system with Kinect. The system allows the user to specify the query motion by performing it directly. Besides, the user interacts with the retrieval system interactively using gestures so no controller is needed and the system delivers a natural user interface.

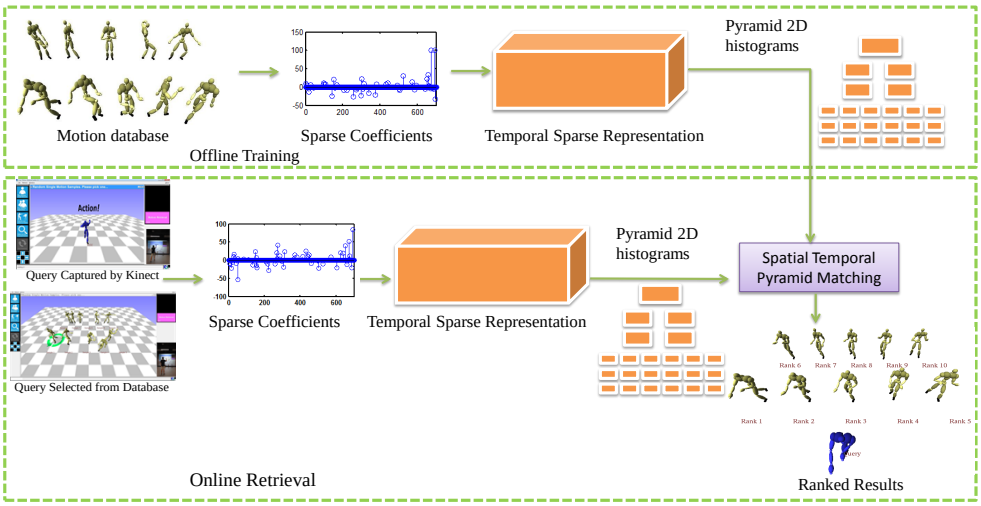


Figure 3.1: Framework of the proposed TSR based motion retrieval system.

3.1 Overview

The overview of the proposed TSR based motion retrieval framework is shown in Figure 3.1, which consists of two major stages: offline training and online retrieval. In the offline stage, all the motions in database are used to train a dictionary by dictionary learning (Section 3.3.1). The sparse representation of each motion can be obtained with the learned dictionary. Meanwhile, temporal sparse representation (TSR) is calculated based one the sparse representation of each motion by considering the temporal information within one motion (Section 3.3.2). TSR is transformed into pyramid 2D histograms by dividing the TSR in different spatial and temporal scales

(Section 3.4.1). In the online stage, pyramid 2D histograms can be obtained for the query motion with the same procedure as offline stage, where the query motion can either choose from existing database or performed by the user directly. Finally, human motion retrieval is conducted with the proposed spatial temporal pyramid matching kernel (Section 3.4.2), which shows the effectiveness in our later experiments.

3.2 Motion Representation

In this chapter, human motion is represented in the Euclidean space. Each human motion sequence is represented as:

$$Y = \{y_1, y_2, ..., y_t\} \tag{3.1}$$

where t denotes the number of frames. And $y_t \in R^{3\times J}$ represents the joint positions of one pose (frame) in 3D space with a total of J joints. The skeleton structure used in this chapter consists of 20 joints as shown in Figure 3.2. These 20 joints are the major joints of the human body to articulate motion. It is consistent with the skeleton definition of Kinect [88]. The motion clips in the database are performed by different users, so the bone length may vary for different performers. To make it comparable between different users with the joint position representation, we adopt a simple yet effective method proposed by Shum et al. [22] to retarget the motion data to a fixed skeleton structure. Given the joint position p^i of joint i and p^j of joint j, where joint j is the parent joint of joint i in the hierarchy structure of the skeleton,

the normalized vector direction between these two joints can be expressed as:

$$u^i = \frac{p^i - p^j}{|p^i - p^j|} \tag{3.2}$$

The retargeted position of joint i can be calculated as:

$$p_r{}^i = p_r{}^j + u^i \times D(i,j) \tag{3.3}$$

where $p_r{}^j$ is the retargeted position of joint j, $D(i,j)$ is the bone length of joint i and j for the commonly used skeleton. As we can see, the retargeted position depends on its parent joint, thus it should start with the top level joint, namely the root joint (HIP_CENTER). Here, we conduct the other two steps for normalization of the motions in database to make it invariant to the body orientation and location of the movement: 1). Remove the global 3D translation information by translating the root joint to the origin of the coordinate system; 2). Remove the global rotation along the vertical axis.

3.3 Temporal Sparse Representation

Sparse coding has been successfully used for image denoising [16], face recognition [54] and human action recognition [55] etc. Sparse coding (SC) and vector quantization (VQ) are highly related. Traditional VQ method applies K-means clustering to find K cluster centers, which are called codebooks. Each sample vector in the database is assigned to one and only one of the centers in terms of the minimum Euclidean

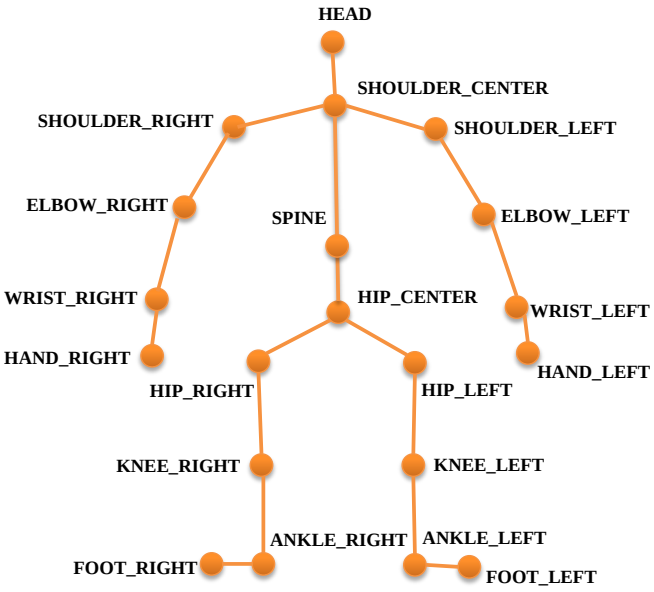


Figure 3.2: Commonly used set of joints for all the motions in database.

distance. This VQ procedure can be viewed as a special case of sparse coding, where each sample vector is represented by only one cluster center. Such representation is a coarse approximation. However, we can relax this constraint to reduce the approximation errors, where each sample vector is represented as a linear combination of the atoms (cluster center in VQ). A set of atoms together form the dictionary in sparse coding. The corresponding coefficients are regarded as sparse representation of the original input data, which has been verified as effective and discriminative for image based applications. Human motion capture data can be considered as a sequence of images, where each image represents one pose. When traditional SC is applied to human motion, each pose corresponds to its own sparse coefficients. However, human motion is a time series sequence and traditional SC neglects the temporal information within motions. To capture the temporal information within one motion, we propose a novel temporal sparse representation (TSR) by considering the relations between

the sparse coefficients.

3.3.1 Dictionary Learning

The dictionary learning procedure can be interpreted as the process of obtaining a set of atoms that can be used to approximate the data through linear combinations. It can be represented as the following optimization problem:

$$\min_{D} \{||Y_A - DX_A||^2\} \quad s.t. \ ||x_i||_0 \ \leq \ m, \tag{3.4}$$

where $Y_A = \{y_i\}_{i=1}^{n_A}, y_i \in R^d$. Y_A is the concatenated matrix of All the motions in database, and y_i is a column vector that represents each frame of the motions. d is the dimension of each frame, which is 60 in this chapter. $D \in R^{d \times q}(q > d)$ is the dictionary to be learned. $X_A = \{x_i\}_{i=1}^{n_A}, x_i \in R^q$ is the sparse representation of Y_A over the dictionary D. $||x_i||_0$ is the L_0-norm that counts the number of nonzero elements in x_i. It has been demonstrated that obtaining the exact sparsest representations is an NP-hard problem [89]. Thus approximate solutions are proposed to solve the problem in the past decades. Efficient pursuit algorithm based on greedy strategy is one direction for approximation, such as the matching pursuit (MP) [90] and the orthogonal matching (OMP) [91] algorithms. Another well-known approach is the Basis Pursuit (BP) [92], which also solves the problem by replacing the L_0 term with L_1 so as to transform the problem to be a convex one. Recently, Aharon et al. [93] developed a method called K-SVD to solve the problem. K-SVD performs two steps to solve the optimization problem: 1). sparse coding and 2). dictionary update. In

the first step, the dictionary is fixed, and X_A is computed by the pursuit algorithm OMP. In the second step, SVD is used to decompose the residual matrix, allowing the atoms of D and the relevant components in X_A updated sequentially. For more details about K-SVD please refer to [93].

3.3.2 TSR Encoding

Given one motion Y, $Y = \{y_i\}_{i=1}^{t}, y_i \in R^d$, the corresponding sparse representation X can be obtained over the learned dictionary D, where $X = \{x_i\}_{i=1}^{t}$, t is the number of frames for this motion. $x_i \in R^q$, and q is the size of the learned dictionary D. x_i represents the contribution of the atoms in the dictionary that are used to approximate y_i. The procedure of obtaining sparse representation X can be formalized as the following optimization problem:

$$\min_{X} \{||Y - DX||^2\} \quad s.t. \ ||x_i||_0 \ \leq \ m, \tag{3.5}$$

The OMP algorithm is adopted to solve the above equation as [93] suggested. It can be observed from the above equation that the temporal information between frames is not considered in obtaining the sparse representation X. In this chapter, we consider the temporal information based on the sparse representation by calculating the relationships between frames of sparse representation. More specifically, we calculate the outer product between two frames of sparse representation. The two frames are chronologically chosen based on the temporal order. In our initial experiment, we calculated the outer product between two adjacent frames of sparse representation to

capture the temporal information. However, adjacent frames are quite similar due to the high frame rate. Therefore, the outer product between two adjacent frames cannot capture much temporal information as these two fames are too similar. Hence, we propose to use a gap value of frame index to calculate the relationship of two frames of sparse representation. More specifically, we calculate the outer product between frame i and frame $(i+\sigma)$ of the sparse representation, where σ is the gap value of frame index. The temporal sparse representation (TSR) can be formulated as follows:

$$TSR = [s_1, s_2, ..., s_m] \tag{3.6}$$

where

$$s_i = x_i x_j^T = \begin{bmatrix} u_1 \\ u_2 \\ ... \\ u_d \end{bmatrix} \begin{bmatrix} v_1 & v_2 & ... & v_d \end{bmatrix} = \begin{bmatrix} u_1v_1 & u_1v_2 & ... & u_1v_d \\ u_2v_1 & u_2v_2 & ... & u_2v_d \\ ... & ... & ... & ... \\ u_dv_1 & u_dv_2 & ... & u_dv_d \end{bmatrix} \tag{3.7}$$

$$i = \{1, 1+\sigma, 1+2\sigma, 1+3\sigma, ..., 1+k\sigma\} \tag{3.8}$$

$$j = \begin{cases} i+\sigma, & \text{if}(i+\sigma < t) \\ f_m, & \text{if}(i+\sigma \geq t) \end{cases} \tag{3.9}$$

In the Equation (3.8), $k = \arg\max_k \{1+k\sigma < t\}$, t is the number of frames for this motion. The entries of sparse representation represent the activation of atoms in the learned dictionary. From the above equation, the outer product between two frames

of sparse representation implicitly considers the relations between the coefficients of the activation of atoms. A toy example of this procedure is illustrated in Figure 3.3. As we can see, the resulted TSR is a $q \times q \times (k+1)$ spatio-temporal volume. Each slice is a square matrix with dimension $q \times q$, and represents the spatial space of TSR. The entry of each slice is the product of the entries of the corresponding sparse representations. The length of TSR is $(k+1)$ which indicates the temporal dimension of TSR. It is not easy to compare between TSRs directly as they are three dimensional arrays. In the next section, we propose a spatial temporal pyramid matching method to do the comparison between TSRs.

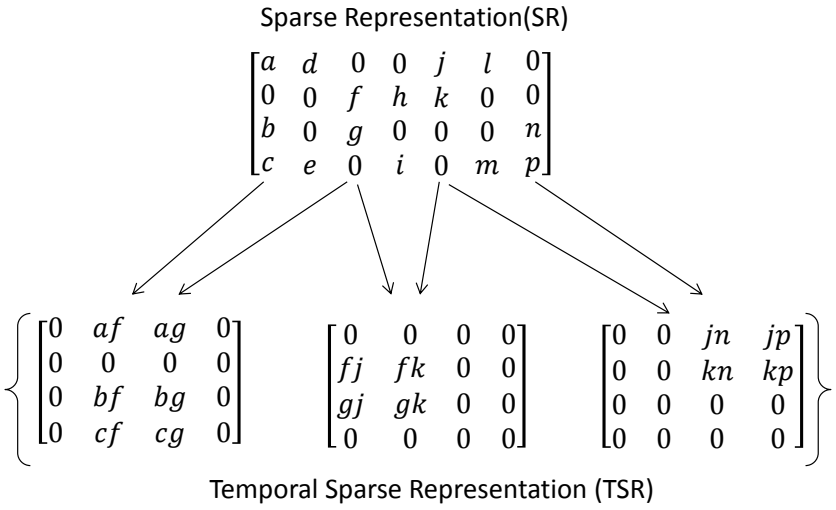


Figure 3.3: Toy example of Temporal Sparse Representation calculation. The top part is the sparse representation with dimension q = 4 and number of frames $t = 7$. The bottom part is the resulted TSR, which is a $4 \times 4 \times 3$ array. In this case, $\sigma = 2$.

3.4 Spatial Temporal Pyramid Matching

As introduced by [94], spatial pyramid matching is proposed as an extension of order-less bag-of-features image representation. Recently, [95] proposed temporal pyramid

matching by adapting the pyramid match kernel to 1-D temporal space. For human motions, variations exist between similar motions both in postures and speed. Similar motions are not necessarily matched numerically in the spatiotemporal domain. Inspired by these works, we propose a method called Spatial Temporal Pyramid Matching (STPM) to address this problem, where the TSR is divided into layers of finer segments in both spatial and temporal domains. This procedure is illustrated on the left side of Figure 3.4. In each level, the TSR is uniformly divided into 3D cubes in both spatial and temporal domains. More specifically, at level l (l = 0, 1, 2,...), it is divided into 2^l segments in temporal domain and $(l+1)^2$ segments in spatial domain. For example, at level $l = 1$, TSR is divided into 2 segments in temporal domain and 4 segments in spatial domain. As a result, there will be $2^l \times (l+1)^2 = 8$ cubes at level l.

3.4.1 Pyramid 2D Histogram Representation

The above procedure allows us to capture the local statistics as the TSR is divided into small finer segments, where each small segment will preserve the local information of the motion. We obtain the final feature representation by a pooling function that can capture the global statistics. Different pooling functions capture different statistics, for example, Yang et al. [15] defined the pooling function as max pooling function on the absolute sparse codes, which was proved to be effective for image classification. In this chapter, we also use the max pooling function to pool over the absolute entries of TSR across temporal domain. The pooling procedure allows the system to capture similar characteristic when the user performs the same motion but with different

speed. We denote the resulting features as pyramid 2D histogram representation. It can be formulated as :

$$Z_{lh}(i,j) = max\{|TSR_1^{lh}(i,j)|, |TSR_2^{lh}(i,j)|, ..., |TSR_t^{lh}(i,j)|\} \tag{3.10}$$

At level l, there are $h = 2^l \times (l+1)^2$ cubes and t is the number of slices corresponding to that cube. $Z_{lh} \in R^{p \times p}$, p is the dimension of the spatial domain of the resulted TSR. For example, at level l, we divide the TSR into $(l+1)$ segments in spatial domain. The dimension of spatial domain is q, thus, $p = \frac{q}{l+1}$. It can be observed that the max pooling function used in this chapter differs from previous works. Here, max pooling is applied across the temporal domain in a 2D matrix. The pooling direction of max pooling function is depicted by the red arrow in Figure 3.4. The resulting feature Z_{lh} can be considered as a 2D histogram and thus we derive the pyramid 2D histogram representation, Z. The similarity between two motions can be obtained by the pairwise comparison between their final feature representation Z at different levels.

3.4.2 STPM Kernel

Equipped with the above information, we further derive our Spatial Temporal Pyramid Matching (STPM) kernel as the similarity matrix the matching between two TSRs. In fact, it can be formulated as a weighted sum kernel:

$$K(Z, \widetilde{Z}) = \sum_{l=0}^{L} \frac{1}{2^l} \sum_{h=1}^{2^l \times (l+1)^2} k(Z_{lh}, \widetilde{Z}_{lh}) \tag{3.11}$$

The above equation indicates that the weights applied in each level ensure that the similarity between smaller cubes plays a less important role. $k(\cdot,\cdot)$ is a kernel that compares the similarity between 2D histograms. In this chapter, we adopt the histogram intersection kernel that shows effective results in our motion retrieval system.

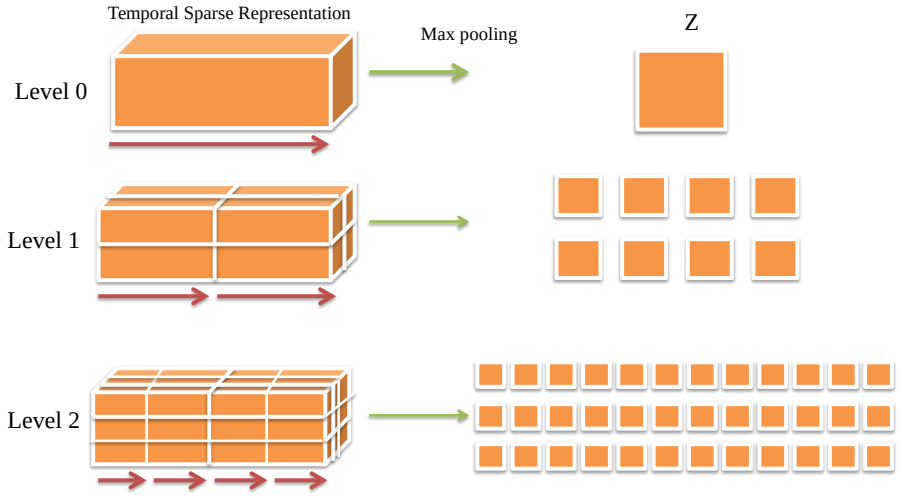


Figure 3.4: The illustration architecture of our Spatial Temporal Pyramid Matching based on Temporal Sparse Representation. At each level, the TSR is divided into segments in both spatial and temporal domains except for level 0. Max pooling function is applied on each obtained cube to get the global statistics. The red arrow represents the pooling direction of max pooling function.

3.5 Controller-free Motion Retrieval System

In this section, we will describe our controller-free natural user interface for motion retrieval. Thanks to the human motion recognition technology derived from Kinect [37], we are able to capture the user's motion as a query. Besides, we use different gestures to represent different control commands, which are more convenient than traditional input devices such as mouse and keyboard. Our motion retrieval system provides two modes for motion retrieval. In the first mode, the user can select the

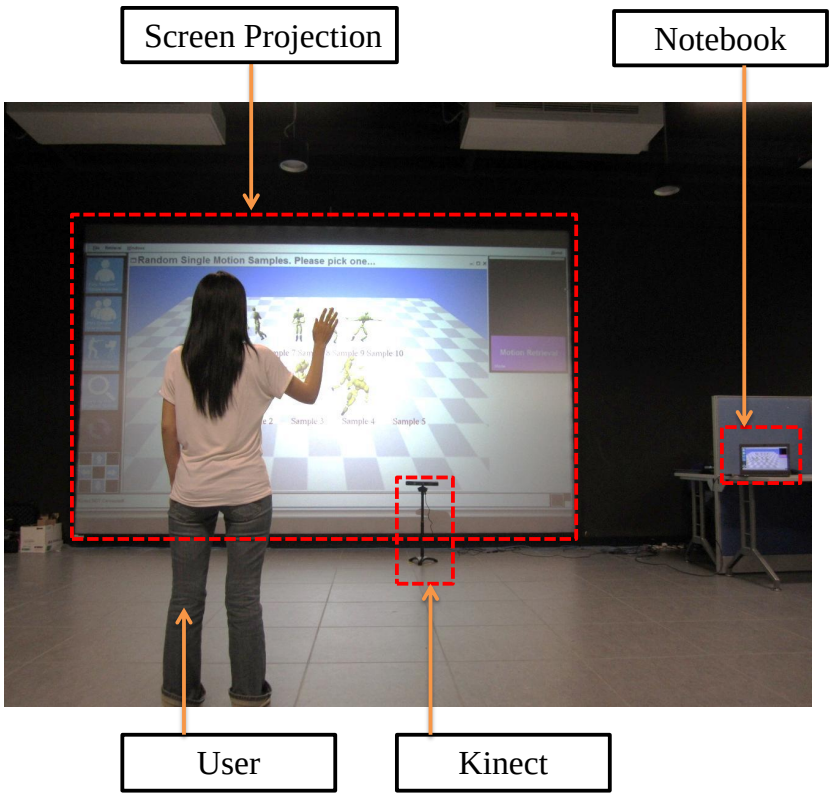


Figure 3.5: The set up of our controller free motion retrieval system.

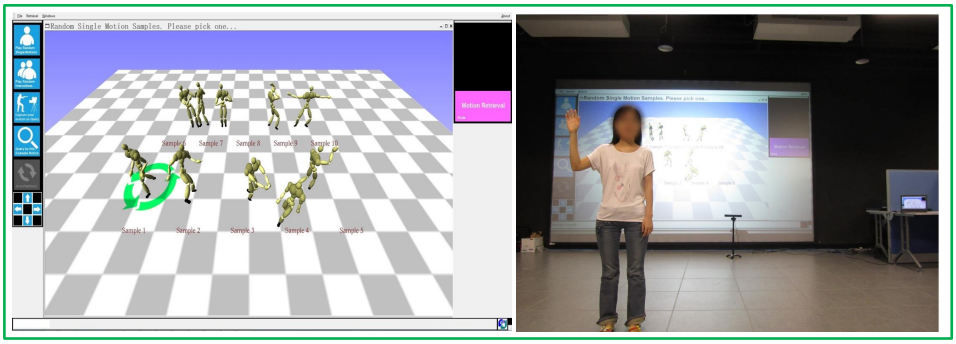
query motion from the motion database as shown in Figure 3.11. The interface displays the motions in database with the user's right hand up. If the user is not satisfied with the motions shown in the current scene, he/she can wave his right hand right and left to view other motions in the next page or the previous page. In the second mode, the retrieval procedure is processed using live captured motion performed by the user with Kinect. It is particularly helpful when the user does not have query motion on hand. It is intuitive to ask the user to perform the query motion, since the user knows what motions he would like to retrieve. The motion captured by Kinect is noisy because of the self-occlusion problem, which will be solved in Chapter 5. However, we use it as a query motion to retrieve more accurate motions from motion capture database. The user can switch to the second mode with his hand clapping as

shown in Figure 3.6(b). The user performs a motion he wants, and then our system records his movement and considers it as a query motion to retrieve similar motions in the database.

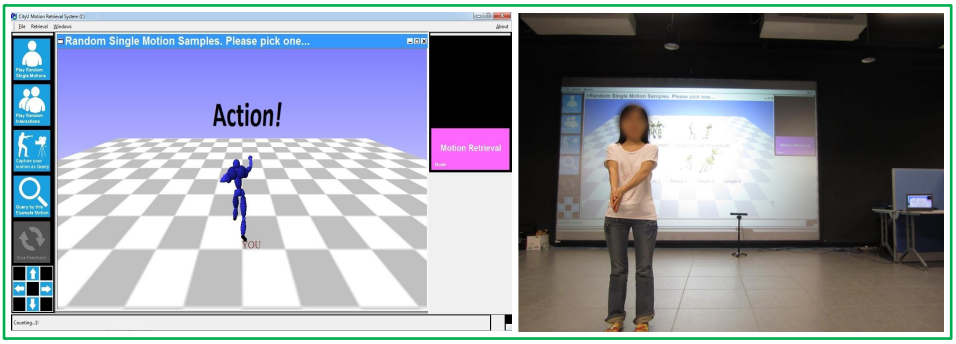
There are other gesture commands in our system. For example, the user can zoom in/out the scene with the distance changing between hands as shown in Figure 3.6(c). The rotation of the view point can be made via the rotation of the user's right hand as shown in Figure 3.6(d).

3.6 Experimental Results

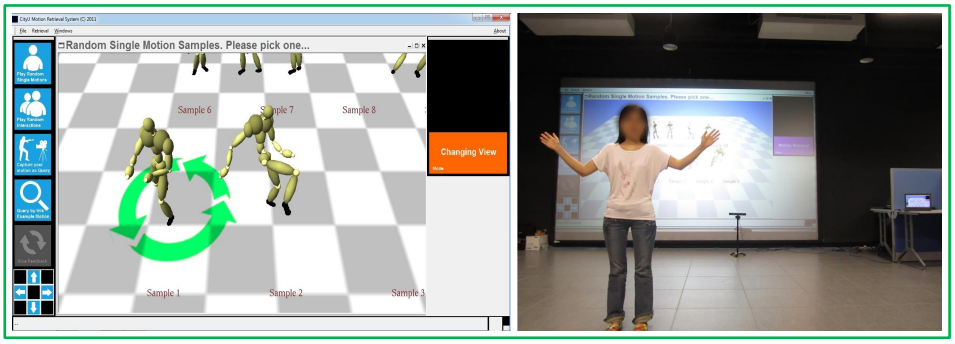
In this section, we evaluate the proposed method for human motion retrieval. All the experiments were conducted on a desktop computer with Intel Core 2 Duo 3.17 GHZ processor. The data set used was obtained from the public HDM05 database [10]. The HDM05 database consists of 130 different motion classes, with multiple trials performed by five subjects in each class. Among the 130 classes of motions, some of them are quite similar to each other and some motions contain more than one action. To better compare our method with previous works (i.e. [5–7]), we chose 10 different types of human motion sequences to form our experimental database. There are in total 210 motion clips, including walking, jumping jack (jumpj), kicking, punching, hopping, sit down chair (schair), elbow-to-knee (etk), clapping, throwing and squatting. On average, the search time per query is 6ms. In the following parts of this section, we conducted various experiments to verify the effectiveness of the proposed human motion retrieval scheme.



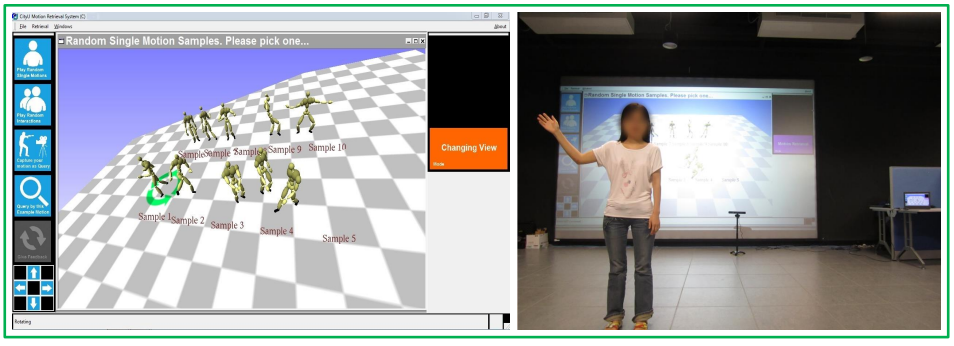
(a)



(b)



(c)



(d)

Figure 3.6: The interface and some example functions of our motion retrieval system. a) Right hand up to show the motions in database; b) Clap hand to capture the user's motion as a query; c) Both hands to zoom in the scene; d) Rotating right arm to rotate the view point.

3.6.1 Parameter Settings

The selection of parameters for dictionary training, TSR and STPM will be detailed in this section. It should be noted that we set the parameters based on the query data from existing motion capture database without using the query motion captured by Kinect as the motion capture data is more accurate and robust.

Dictionary Training Parameters. There are several parameters in our motion retrieval mechanism. In the dictionary learning stage, all the motions in database are concatenated together to be used as the training data, where each column corresponds to one frame. As presented in section 3.2, joint position is used as the motion representation. A totally of 20 joints are chosen from the hierarchy skeleton, meaning that the dimension of each frame is 60. In the sparse coding stage, the number of non-zero elements of sparse coefficients and the size of the dictionary are empirically set to be 35 and 700 respectively. In considering these two parameters, it is important to balance the computation efficiency and the reconstruction error induced in Equation (3.5). To achieve a smaller reconstruction error of the objective function, more computational time will be required and vice versa.

TSR and STPM Parameters. The gap value σ between frame index can be regarded as a down sampling parameter. The larger the value of σ, the larger will be the interval between frames, meaning that less frames of sparse representation will be extracted for TSR calculation. TSR is based on the outer product between two frames of sparse representation. At the same time, when the value of σ is large, it may not be able to capture the temporal variations during the time gap. On the

other hand, two frames tend to be similar if the value of σ is small since their indices are close, thus it will not contain much temporal information when the chosen two frames are very close in time.

To capture the feature of TSR with various levels of detail, we derived STPM by dividing TSR in both spatial and temporal domains at different scales with l levels. The top level captures more global statistics of the motion while the lower levels contain finer details but are also more prone to noise thus it is essential to determine a suitable number of levels.

As explained above, the selected values of l and σ will influence the retrieval results. We chose these two parameters alternatively as they are independent. More specifically, we fix one parameter and choose the other one based on the optimal average retrieval accuracy. The retrieval accuracy calculation will be presented in Section 3.6.2. The relationships between these two parameters and retrieval accuracy is shown in Figure 3.7. From the results, we can observe that the retrieval accuracy is less than 0.8 without the usage of pyramid matching ($l = 1$). For each selection of σ, the system achieves the best result when $l = 3$. The best retrieval accuracy can be obtained when $l = 3$ and $\sigma = 7$. The result is consistent with our expectation. On the other hand, the curve (retrieval accuracy) tends to be flat if the values of l and σ located in a small range. It means that our method is robust and not that sensitive to the selection of the two values.

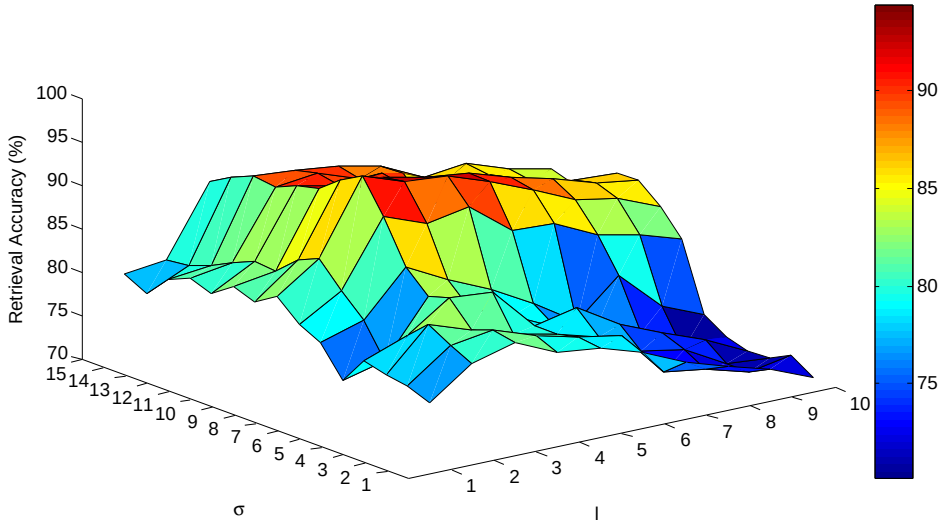


Figure 3.7: The retrieval accuracy between two parameters l and σ. l axis represents the level value and σ axis represents the gap value.

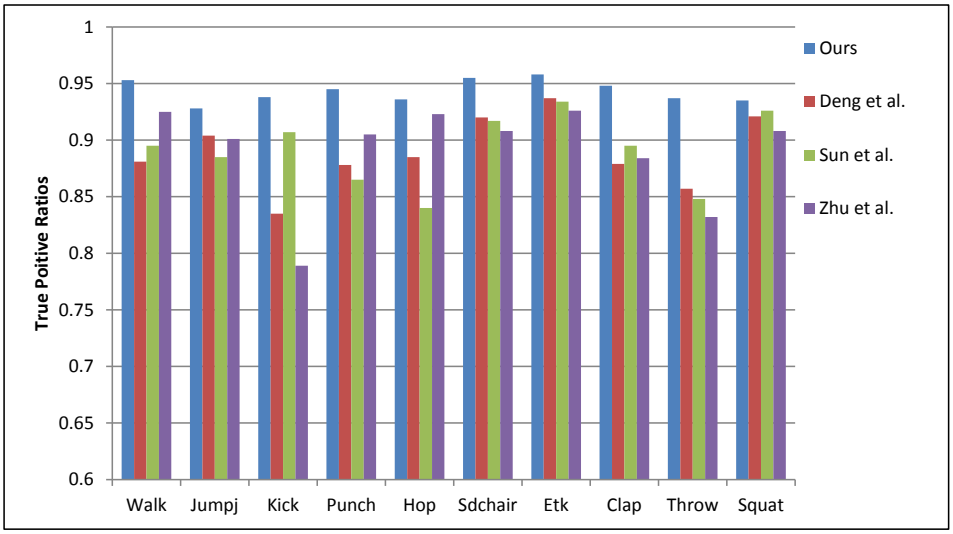
3.6.2 Performance Evaluation

In this section, we compared our human motion retrieval system with three other related works [5–7]. In [5], Deng at al. transformed motion retrieval into string matching, which is based on the hierarchical motion representation. In [6], Sun at al. conducted motion retrieval using low-rank subspace decomposition of motion volume, where motion volume is constructed from the self-similarity matrix of each motion. In [7], Zhu at al. adopted sparse coding in quaternion space, where each motion is represented as one dictionary, the similarity between motions is computed with the optimal one-to-one map between corresponding dictionaries by minimizing the total distance. In the following sections, we will detail the comparison between our approach and these three related works.

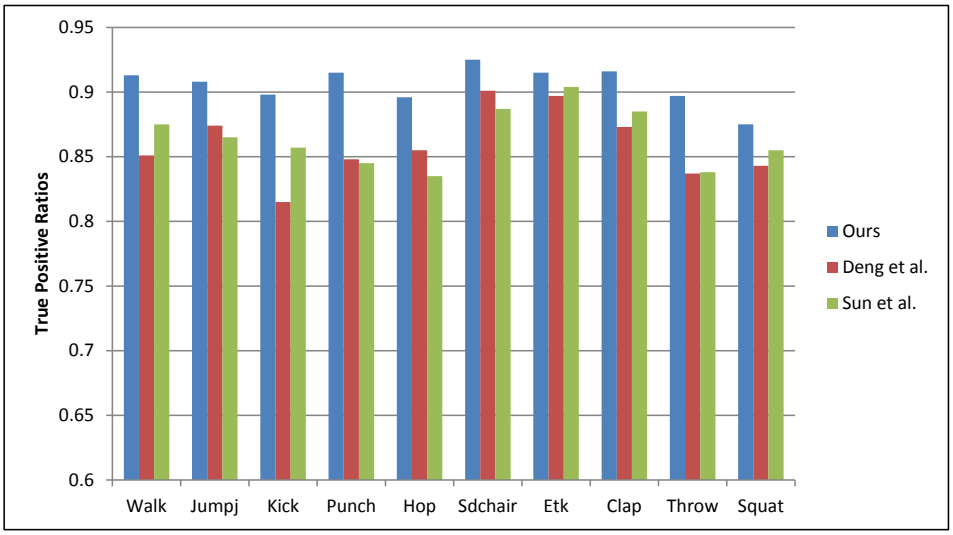
Retrieval Accuracy. To evaluate the accuracy of the proposed approach for human motion retrieval, we adopt a similar strategy as Kovar et al. [49]. The objective

of motion retrieval is to retrieve similar motions that belong to the same class as the query motion. Here we use two types of data sets: one is called single type (ST) data set and the other is called mixed types (MTs). ST consists of motions of the same class. MTs consist of multiple classes of motions. Given a query motion, we retrieve motions from these two data sets, where ST is the same class as the query motion. The results from ST data set are considered to be the ground truth. We collect four classes of motions as ST data sets, including walking, kicking, jumping jack and punching. True positive (TP) is used as the accuracy criteria to evaluate the retrieval results, which is defined as the percentage of correctly retrieved results from MTs that are in the retrieval results from ST, where ST is with the same category as the query motion. Here, we use the top 20 results for TP calculation. As our system provides two modes for the query specification, here, we also evaluate the accuracy performance for these two modes separately. For the first mode, we choose the query from the database directly. For the second mode, we ask the user to perform the query motion to be captured by Kinect. We did not compare with [7] in the second mode since the approach of [7] is based on quaternion whereas the data captured by Kinect are joint positions. Figure 3.8(a) and Figure 3.8(b) demonstrate that our approach outperforms the other three methods. For the first mode, the average true-positive ratio of our method is 0.947 and 0.889, 0.892, 0.895 for three other compared methods respectively. For the second mode, the average true-positive ratio of our method is 0.906 and 0.859, 0.864 for the other two compared methods respectively.

Confusion Matrix. We not only consider the correctly retrieved results with true positive ratios, but also evaluate the mistakenly retrieved results using a con-



(a)

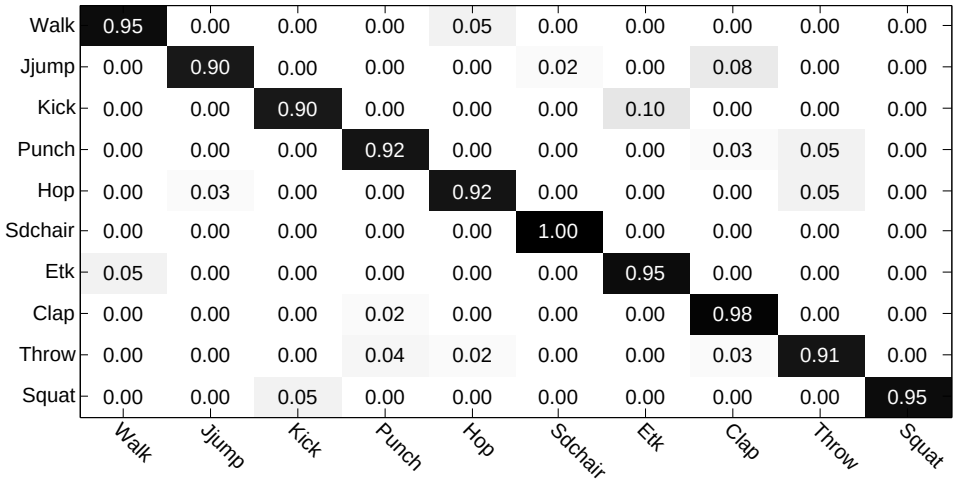


(b)

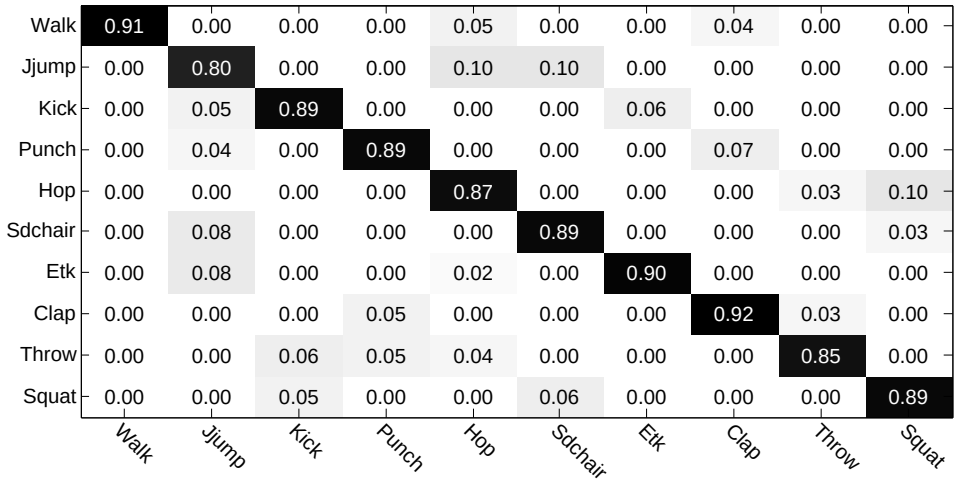
Figure 3.8: True positive ratios of the proposed method, Deng et al. [5], Sun et al. [6] and Zhu et al. [7]. (a) Query selected from existing motions; (b) Query captured by Kinect.

fusion matrix. Confusion matrix is a specific table layout which can visualize the performance of classification algorithm, where each entry reveals the confusion between two classes. Here, we use confusion matrix to evaluate the confusion of retrieved results, where each column represents the instances in a retrieved class and each row represents instances in the actual class. As a result, it is easy to tell whether the retrieval system is confusing two classes. Figure 3.9(a) and Figure 3.9(b) are the confusion matrix for these two modes respectively. As we can see, it can differentiate motions in various classes very well. Only some classes are confused. For example, punch motion is confused with clap motion and throw motion. It is acceptable as these three motions preserve similar patterns among them although they belong to different types of motions.

Precision VS. Recall. In addition to the evaluation of the retrieval accuracy, we also use precision-recall to verify the robustness of our human motion retrieval system. Precision is the ratio of correctly retrieved motions to the total number of retrieved motions. Recall is the ratio of correctly retrieved motions to the total number of relevant motions in the database. The average precision-recall curve is shown in Figure 3.10. Here, we only use the first mode for evaluation, in which the query is selected from the motion capture database. It can be observed that the performance of our approach is much better than the compared methods, which indicates the robustness of the proposed human motion retrieval system.



(a)



(b)

Figure 3.9: Confusion matrix for 10 classes.(a) Query selected from existing motions; (b) Query captured by Kinect.

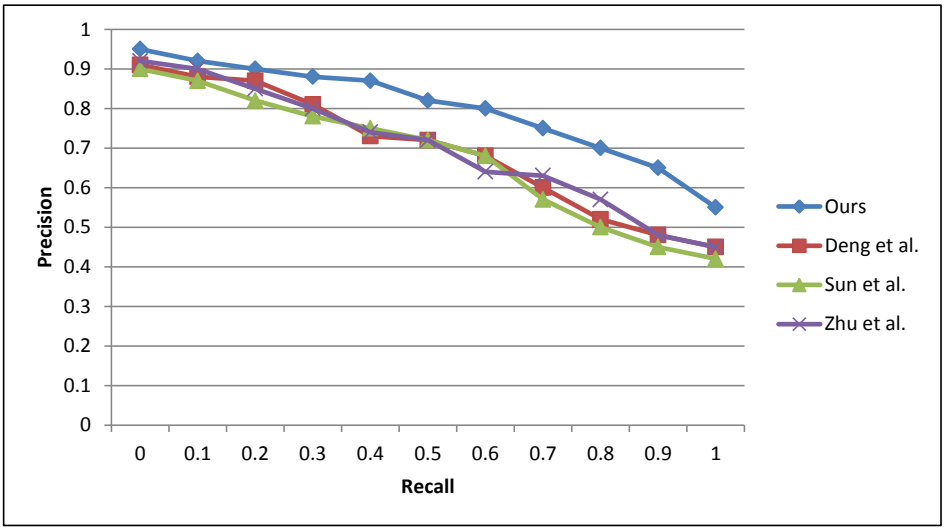


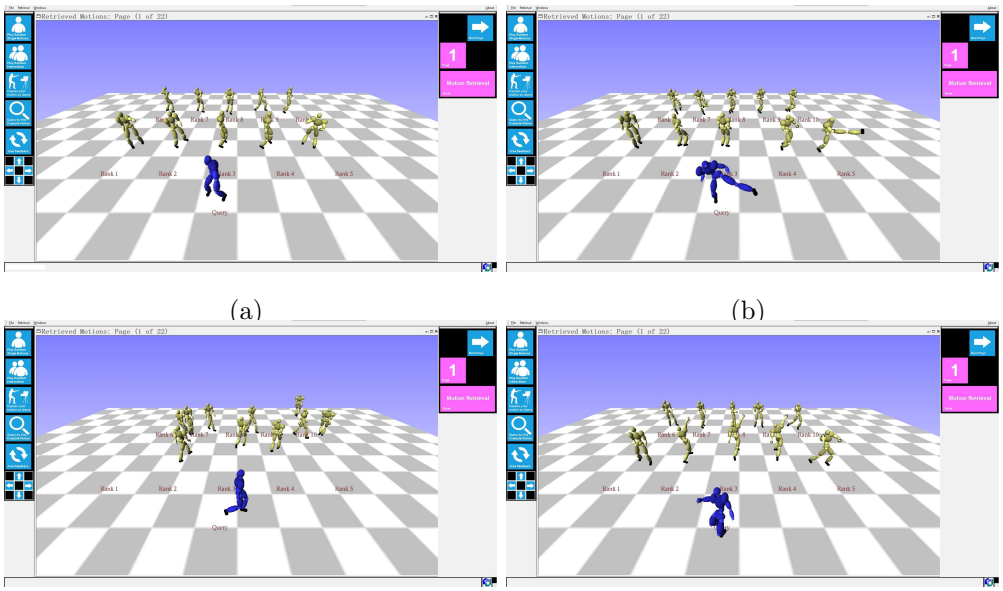
Figure 3.10: The Precision-Recall curves of the proposed method, Deng et al. [5], Sun et al. [6] and Zhu et al. [7].

3.6.3 Demonstration

We have implemented a prototype of the proposed TSR based human motion retrieval system, in which we have also integrated the gesture commands presented in Section 3.5 into the system. The system allows the user to control it in a more intuitive way. Some results and the interface of the system are shown in Figure 3.11. The buttons of the system allow the user to select the query motion from the database and conduct the retrieval procedure. The character in front represents the query motion and others are the ranked retrieved results. From the results we can observe that the top ranked retrieval results are similar as the query motion.

3.7 Summary

In this chapter, we propose a unified framework for human motion retrieval which allows the user to retrieve desired motions from the database in an efficient and effec-



(a) (b)

(c) (d)

Figure 3.11: The interface and retrieval results of the proposed human motion retrieval system. a) Punch motion; b) Kick motion; c) Walk motion; d) Throw motion.

tive way. To this end, we propose spatial temporal pyramid matching that captures both the global and local statistics based on the proposed temporal sparse representation. Unlike the traditional sparse representation, our temporal sparse representation encodes the temporal information by considering the relationship between frames of sparse representation. In our system, a natural interface is developed with Kinect, and the system provides two modes for query specification for human motion retrieval. One mode allows the user to select the query as an example motion from database. The other mode allows the user to perform a query motion by himself, which is live captured by Kinect. Besides, our Kinect based retrieval system allows the user to control the interface with gestures, such as motion selection with hand waving, zoom in/out the interface through varying the distance between hands etc.

Chapter 4

Human Motion Variation Synthesis

Human motion variation synthesis is important for crowd simulation and interactive applications to enhance the animation quality. With the proposed method in Chapter 3, we can retrieve similar motions from the motion capture database and then use the retrieved results for crowd simulation. However, what if there are not enough similar motions for an intended application? Can we automatically synthesize a group of similar motions? To solve this problem, we propose a novel generative probabilistic model to synthesize variations of human motion. The key idea is to model the conditional distribution of each joint via a multivariate Gaussian Process model, namely Semiparametric Latent Factor Model (SLFM). SLFM can effectively model the correlations between degrees of freedom (DOFs) of joints rather than dealing each DOF separately as implemented in existing methods. Detailed evaluations are conducted to show that proposed approach can effectively synthesize variations of different types of motions. Motions generated by our method show a richer variations compared to existing ones. Finally, the user study shows that the synthesized motion has similar

level of naturalness as motion capture data. Our method has great potential to be applied in computer games and animations to improve the quality of animations by introducing motion variations.

4.1 Motion Representation

In Chapter 3, we represented human motion in the Euclidean space with 3D positions. In this chapter, the hierarchy structure of each frame is represented as root positions and joint rotations. The reason will be detailed in Section 4.2.2. The representation of each frame at time t is formulated as:

$$y_t = \{p_0, q_0, q_1, ..., q_i\} \tag{4.1}$$

where p_0 and q_0 are the global world 3D positions and orientations of the root joint, q_i is the rotation of the ith joint with respect to its parent joint. We encode the joint angles with parameterized exponential maps [96]. The joints are highly correlated during movements based on the articulated skeleton structure. Dividing the skeleton into partitions, either to reduce the complexity for motion recognition [97] or for partial motion synthesis to enrich motion database [31] [65], has been effective. We therefore divide the human skeleton into five partitions based on the skeleton hierarchy as shown in Figure 4.1, which are RA (Right Arm), LA (Left Arm), RL (Right Leg), LL (Left Leg) and TH (Torso and Head). There are two reasons for us to partition the skeleton. First, it helps us to extract the relations between joints as the

joints are more correlated within the same body part (e.g. wrist, elbow, shoulder). Second, we only need to take into account the joints that are in the same partition when we considering one joint, which makes the system simpler.

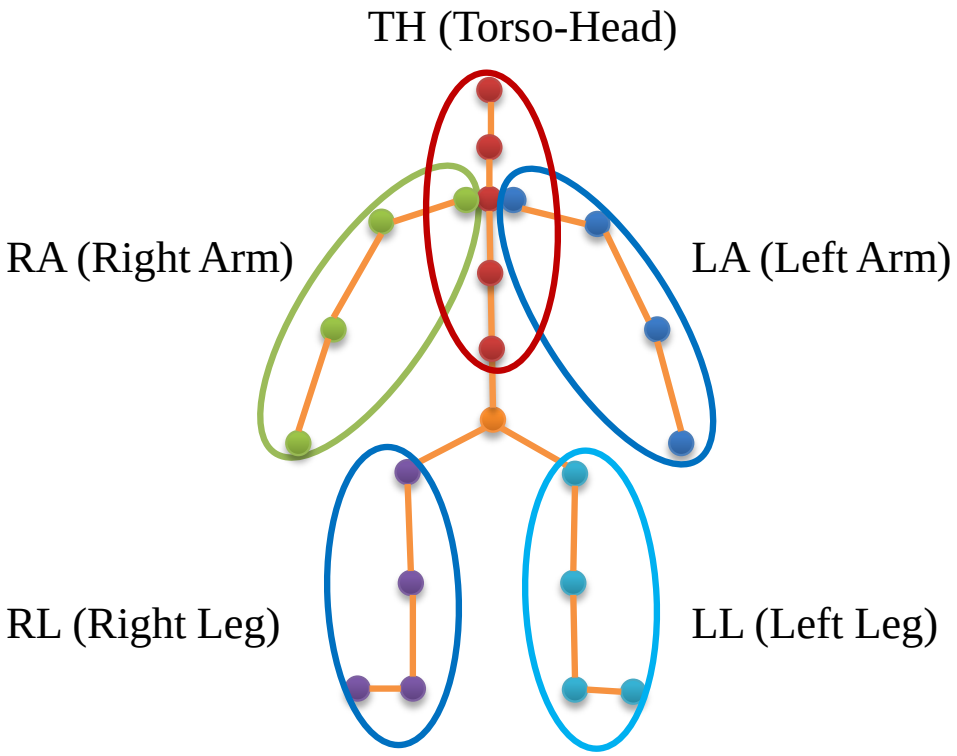


Figure 4.1: Five partitions of the skeleton, RA (Right Arm), LA (Left Arm), RL (Right Leg), LL (Left Leg) and TH (Torso and Head). The same color dots represent joints within the same partition.

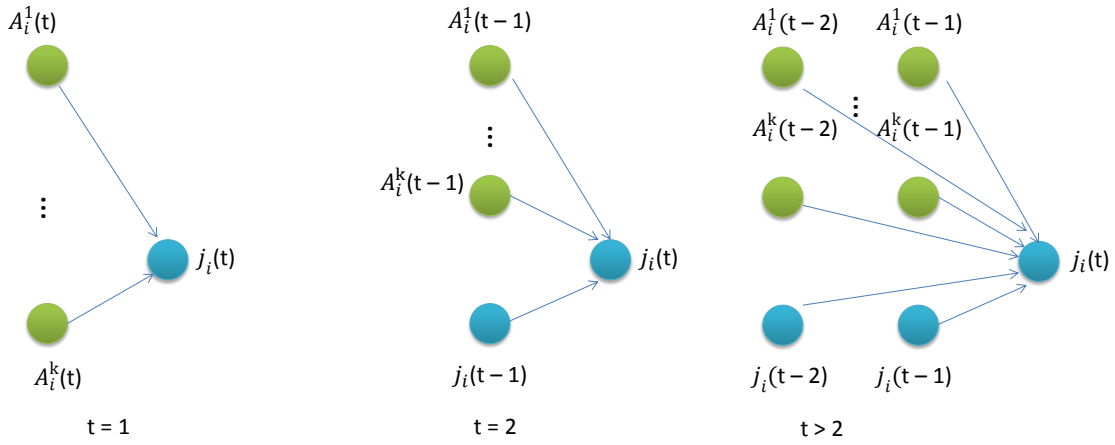
4.2 Model Construction

In this section, we describe the way of specifying the dependency between joints with our partition based skeleton structure and the features used in SLFM prediction model.

4.2.1 Partition based Structure

Predicting the distribution of one joint based on some prior knowledge is the most important component for motion variation synthesis. In this work, we extract features from the ancestor joints of the current joint, which are defined as input features, as prior knowledge for prediction. By this way, the relations between joints in the same partition are formalized as the conditional dependency between one joint and its ancestor joints.

For a given joint j_i, its ancestor joints are defined as the set of all joints in the higher levels of the skeleton hierarchy within the same partition as j_i, which are denoted as A_i. Using the right wrist joint as an example, its ancestor joints are right elbow, right shoulder and right clavicle. It should be noted that the top level joint (e.g. right clavicle) has no ancestor joints, thus $A_i = \emptyset$. Figure 4.2 shows the relationships between joint j_i and its ancestor joints. The blue dot represents the given joint j_i at time t, $t-1$ and $t-2$. The green dots are ancestor joints of j_i at time slices $t-1$ and $t-2$, which are denoted as $A_i(t-1)$ and $A_i(t-2)$, respectively. A_i^k represents the k-th joint in A_i. More specifically, A_i is one set of joints and A_i^k is one joint that belongs to A_i. Starting from the third frame, we adopt a second order temporal model to predict the distribution of one joint at the current frame. This is an observation we found in our initial experiments, in which the first order model performs sub-optimally. In particular, by analyzing the synthesized motions, we find that the movements tend to randomly move away from the original input data, since only one previous frame is not enough to constrain movement ranges, so we need to



(a) (b) (c)

Figure 4.2: The graphical representation of the conditional dependency between joints. Blue dot represents joint j_i, green dots represent the ancestor joints of j_i. a) at time $t = 1$; b) at time $t = 2$; c) at time $t > 2$.

consider the information from more frames. To minimize the noise introduced by the newly added information, we have to select suitable features for prediction. In the following part, we will detail the feature selection both for input features and output features.

$$P(j_i[t]) = \begin{cases} P(j_i(t)|J_{pi}(t-1), J_{pi}(t-2), \dot{J}_{pi}(t-1), \dot{J}_{pi}(t-2)), & t > 3; \\ P(j_i(t)|J_{pi}(t-1), J_{pi}(t-2), \dot{J}_{pi}(t-1)), & t = 3; \\ P(j_i(t)|J_{pi}(t-1)), & t = 2; \\ P(j_i(t)|A_i(t)), & t = 1. \end{cases} \tag{4.2}$$

4.2.2 Feature Extraction

To facilitate the calculation of conditional distribution in this section, we use the term *parent joints* J_{pi} to represent these joints that will influence joint j_i. Except A_i, j_i itself will also influence j_i thus $J_{pi} = \{A_i, j_i\}$. The partition based structure

constructed above allows us to determine the conditional dependency between one joint and its parent joints, for example, $P(j_i(t)|J_{pi}(t-1), J_{pi}(t-2))$, where $t > 2$. P represents the conditional probability distribution. The semiparametric model SLFM will be used to model the conditional distribution for each joint. The performance of this model depends heavily on the input features, so it is important to define and select effective features.

In this work, we use the skeleton configuration feature (SCF) [66] as the output feature to reconstruct poses. SCF is represented by joint angle and is parameterized by exponential maps [96]. We do not use joint position as the joint representation, as constraining bone lengths defined in the skeleton hierarchy introduces extra system complexity. It should be noted that the output features of root joint are the translation along the ground-plane, and rotation around the vertical axis of the root relative to the root at last frame. With this information we can reconstruct the movement path by integrating the elapsed time and root transformation. Our output feature differs from that used in Lau et al. [8], in which the frame difference is used as the output of their regression model. However, the adjacent frames of one motion are often very similar due to the high frame sampling rate, which limits the range of movement and hence the variation of the synthesized movement.

SCF is also used as the component of the input features. In addition, dynamic feature is introduced to convey the temporal information between frames. The dynamic feature is defined as the positional velocity between current frame and previous frame. For the joints at the first frame, we only use SCF from its ancestor joints as the input feature. For the second frame of each joint, the input feature only includes

SCF from its parent joints as there is no dynamic feature for the first frame. For the third frame, we use the SCF from the first two frames and the dynamic feature from the second frame as the input feature. Both SCF and dynamic feature from previous two frames are used as the input features for the subsequent frames. After defining source and output features, we use the SLFM to model the distribution of one joint at different time slices as given by equation 4.2, where $\dot{J}_{pi}(t-1)$ and $\dot{J}_{pi}(t-2)$ represent the dynamic feature at time $t-1$, $t-2$, respectively. In the following section, we will show the details of modeling the conditional distribution of one joint by SLFM.

4.3 Computing Conditional Distribution by SLFM

Estimating the conditional probability distributions of the multivariate variable $j_i(t)$ from the input features, i.e. Equation 4.2, is the key component of our synthesis method. Traditional methods model this distribution using a parametric model, such as using a multivariate normal distribution and then optimizing its related means and covariance matrices by maximizing the posterior distribution of training instances. However, these methods involve a lot of parameters and often suffer from local optima.

Nonparametric method, like Gaussian Processes (GP) has been extensively used in animation domain, such as motion editing [66], motion generation [70] and motion transition modeling [98]. GP is based on the assumption that adjacent observations should convey information about each other, and the coupling between observations takes place by means of the covariance matrix. More formally, let $\mathbf{X} = [x_1, \ldots, x_N]^T$ be a matrix representing the input data, which is constructed by concatenating the

extracted source features. Let $\mathbf{y} = [y_1, \ldots, y_N]^T$ denote the output values, which is the DOF of joints. y_i is the corresponding output of the input x_i, $y_i = f(x_i) + \epsilon$ where $\epsilon \sim N(0, \beta^{-1})$ is a noise variable, which is independent for each data point. The shape of function f is determined by the selection of covariance function, in this work we use the following covariance function:

$$k_c(x_i, x_j) = \theta_0 \exp\left(-\frac{\theta_1}{2}\|x_i - x_j\|^2\right) + \theta_2 + \beta^{-1}\delta_{ij} \tag{4.3}$$

where δ_{ij} is Kronecker's delta function and $\Phi = \{\theta_0, \theta_1, \theta_2, \beta\}$ are the hyper-parameters.

When the standard GP is applied to human motion modeling, GPs are independently learnt for each DOF of one joint. However, as the DOFs are highly dependent, it is advisable to take the dependency into account. In this thesis, we use the recently developed multivariate GP model Semiparametric Latent Factor Model (SLFM) [34] to predict the distribution for each joint. In our case, our purpose is to predict the distribution of each joint. In our initial experimental testing, GP was adopted to synthesize variations. However, the movements tended to be ambiguous as GP cannot model the relations between the outputs. The DOFs of each joint are highly correlated, so it is more appropriate to consider them together rather than treating them independently. SLFM also inherits the advantages of GP method, e.g. SLFM can be robustly learned from small training data sets and the parameters of similarity function can be optimized without relying on experimental cross-validation. Moreover, variation trend of mean function can be easily adapted by changing the combination factors of different style kernel functions.

The structure of our SLFM for modeling the correlations between three DOFs of one joint is shown in Figure 4.3. Similar to standard GP, each output y_i with $i \in \{1, 2, 3\}$ is independently generated from its own latent function $f_i(x)$. The difference between standard GP and SLFM is that $f_i(x)$ is a linear mixing of some basic GPs $u_i(x)$, which can capture the dependencies that exist among DOFs. The kernel function of latent function $k_l(x)$ is expressed as:

$$k_l(x, x') = \sum_{p=1}^{3} \phi_{i,p}^2 u_p(x, x'), \tag{4.4}$$

where $u_p(x, x')$ is the kernel function of the p-th GP for the input feature instances x and x', and $\{\phi_{i,1}, \phi_{i,2}, \phi_{i,3}\}$ are the mixing weights for $f_i(x)$. SLFM can be viewed as an augmented GP which models output dependencies by sharing kernel hyper-parameters (i.e. the parameters of $u_p(x, x')$) of basic GPs. As a result, we can still use the same learning and prediction method as GP. The basic training procedure is as follows. First, we use a series of motion of the same motion type as coarse training data. Then, we use the feature extraction method described in Section 4.2.2 to extract SCF and dynamic features as the training input and output pairs of SLFM. Finally, using the conjugate gradient descent [99] methods, we obtain the optimal parameters for SLFM. A review of the inference and learning of GP can be found in [99].

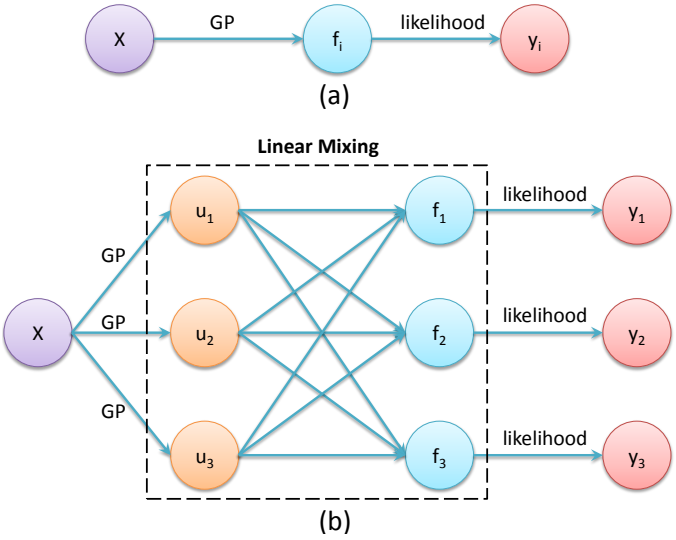


Figure 4.3: The graphical models of (a) standard GP and (b) our used semiparametric latent factor model for three DOFs of a joint. X represents the input features and y_i corresponding to the i-th DOF.

4.4 Motion Synthesis

After obtaining the learned SLFM, we can synthesize a bunch of variations by sampling from the predicted distribution of each joint. Subsequent frames can be iteratively synthesized based on its previous synthesized frames and the learned conditional distribution model. For the joints that have no ancestor joints at the first frame, we sample from the data distribution directly. More specifically, we calculate the mean and standard deviation from the training data of these joints which have no ancestor joints at the first frame. In order to enhance the naturalness of the motions synthesized by our method, we apply the blending technique proposed in [100] to enhance the motion quality. Alternatively, automatic foot strike cleanup methods [101] can be applied to achieve the similar purpose.

4.5 Experimental Results

In this section, we will evaluate the proposed method in different aspects. All experiments were conducted on a desktop computer with Intel Core 2 Duo 3.17 GHz processor. We implemented the system with C++/Matlab. There were two data sets used in our experiments. The first one was the generally used HDM05 [10] database. This database consists of 130 different motion classes, with multiple trials performed by five subjects in each class. The motions were performed at a sampling rate of 120 Hz. We chose three types of motion from this database: walking, single leg hopping and jumping jack. On average, 0.17 second is required to synthesize 1 second of motion for this data set. The second data set included Tai Chi motions captured by an optical motion capture system in our own laboratory. The frame rate is 60 HZ. We chose Tai Chi motion because of its large range of movement and its complicated movement features, which can be used to verify the robustness of the proposed method. The motions were performed 10 times by a professional Tai Chi Master. On average, 0.14 second is required to synthesize 1 second of motion for this data set.

4.5.1 Model Evaluation

For each type of the motion, we choose 10 example motions as training data set to learn SLFM. Figure 4.4 shows the synthesized Tai Chi and jumping jack motions by our method. We can observe that the synthesized motion preserves the major features of the original one, but differs in both spatial and temporal domains. For the Tai Chi motion, the range of movement varied at different phases. For the jumping

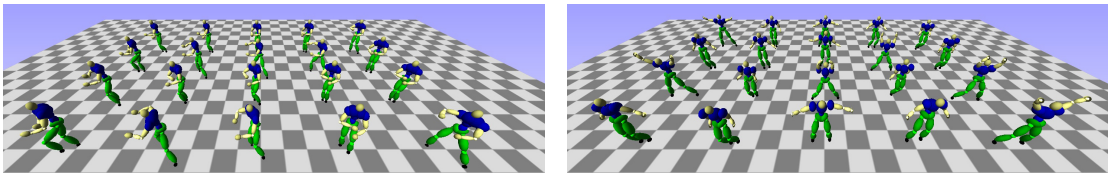


Figure 4.4: Human motion variants synthesized by our approach. Left side: The variants synthesized from Tai Chi motion. Right side: The variants synthesized from jumping jack motion.

jack motion, the heights of swing arms and jumping legs also varied among different variants. The synthesized variants of walking motion had different degrees of arm swing as well as the stride length. For the single leg hopping motion, the maximum height reached by the jumping motion varied across the variants.

To verify the synthesized motion is similar to the input one, we visualized the results by applying Principal Component Analysis (PCA) on both motions. The visualization of the first PCA dimension is depicted in Figure 4.5. We can observe that the synthesized motions (the gray lines) have the similar variation trend as the original input data (the red line), so the motion type of the original input data can be well preserved, while variations are added.

It is difficult to precisely define the degree of variation, and it is also hard to define how many variations are enough for intended applications. Instead of making such a definition, we introduce a method to characterize the variation among motions. Given a set of motions, we use the difference between pairwise motions to evaluate the variations. Specifically, given N motions, we calculate the difference between one motion and the other (N-1) motions, where the difference is calculated by Dynamic Time Warping (DTW) [102]. Hence, we can define the variation among this set of

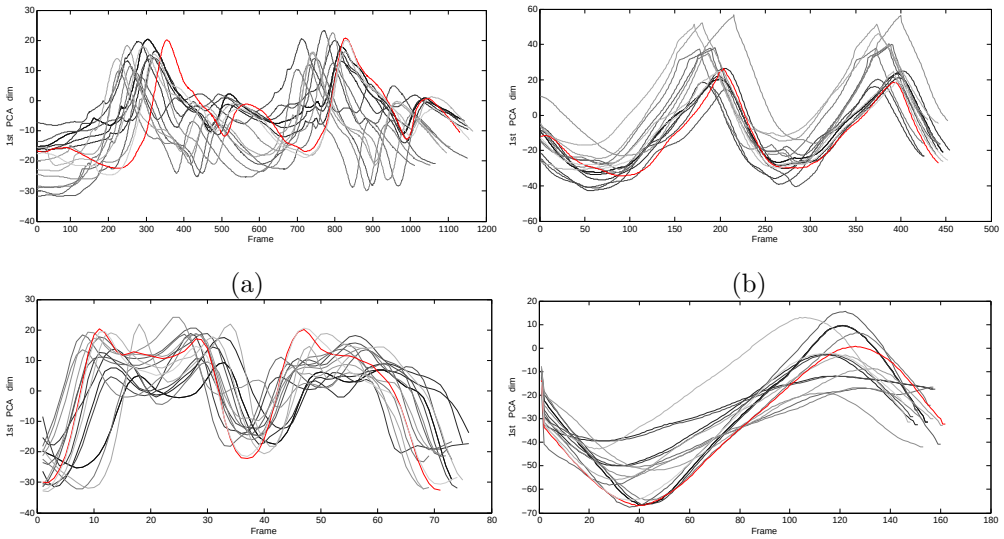


Figure 4.5: Plots of 15 variants and one of the training data. Each curve represents the first PCA dimension of one motion, where the red curve represents one of the training data and others represent the synthesized results. a) Tai Chi motion; b) Walking; c) Single leg hopping; d) Jumping jack.

motions as:

$$V = \sum_{i=1}^{N} \sum_{j=i+1}^{N} \frac{2}{N(N-1)} D_{tw}(M_i, M_j) \tag{4.5}$$

where N is the total number of motions, and D_{tw} is the distance between two motions calculated by DTW. Here, we manually selected a set of motions that show obvious variations, and we calculate the variation with Equation 4.5, which is 435.8. We make an assumption that 435 is the threshold for the user to observe the variations among motions. We also quantize the variation for motions synthesized by our approach and Lau et al. [8], which are 558.7 and 465.4 respectively. It demonstrates that both our method and Lau et al. [8] can synthesize enough variations, whereas our method can synthesize more obvious variations compared with Lau et al. [8] .

To verify that our approach can learn from small training data set, we plot the

number of training data used against the variations synthesized by our approach. Here, we calculate the variations among 20 variants using Equation 4.5. The result is shown in Figure 4.6. We can observe that the degree of variations is increasing greatly with the number of motions increased. The curve tends to be flat when the number of training data is more than 10, which demonstrates that our approach can learn from small training data (i.e. less than 10).

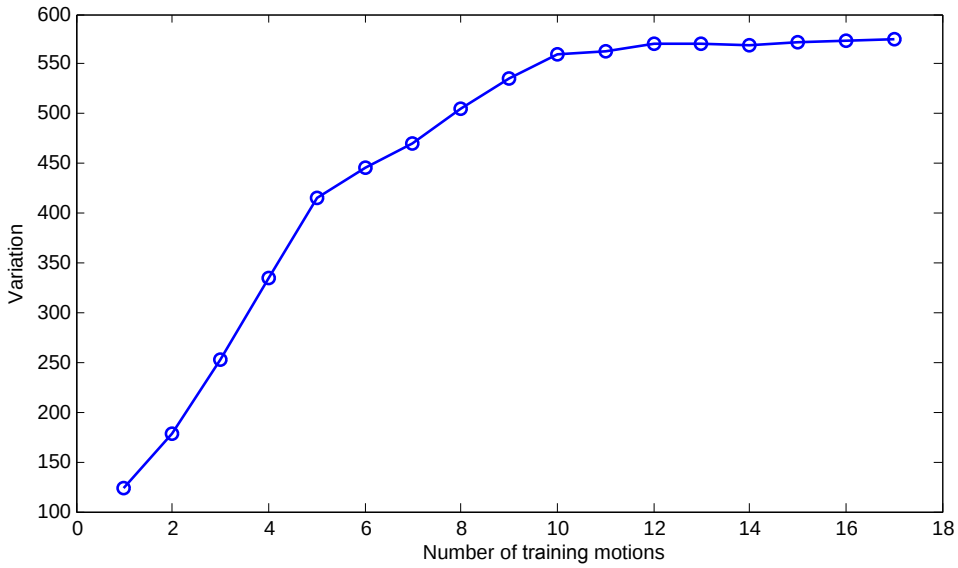


Figure 4.6: The variations synthesized by our approach with different number of training motions.

4.5.2 User Study

To evaluate the naturalness of the motions synthesized by our method, we compared the synthesized motions with motion capture data by conducting a user study evaluation. A total of 10 participants were invited. All of them had little or no experience about motion capture and 3D animation. We created a set of motions consisting of motion capture data and synthesized motion data. Participants were asked to give

a score for each motion based on its naturalness without knowing whether the displayed motion is captured or synthesized. The score ranges from 1 to 10 (inclusive), where 1 means the most unnatural, and 10 means the most natural. We performed a two-way Analysis of variance (ANOVA) [103] on the obtained scores. There are two factors that influence the score, one is motion type (tai chi, walking, hopping, jumping jack) and the other factor is method (motion capture, our method). The result is shown in Table 4.1. The result tells that the user scores are not affected by the motion types and capturing methods. There is no significant difference in the obtained scores in terms of perceptual naturalness between motion capture data and synthesized motions. It verifies that our method produces as natural motions as motion capture data.

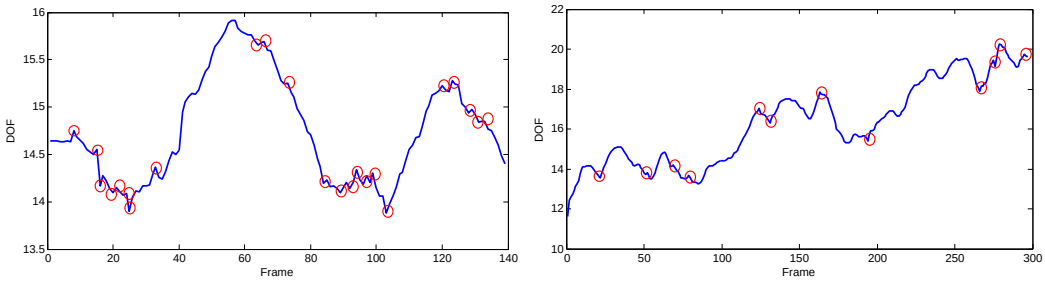
Table 4.1: Two-way analysis result between the naturalness and two factors (method and motion type).

Source	SS	df	MS	F	Prob > F
Motion Type	1.00	3	0.33333	0.67	0.5775
Method	6.45	9	0.71667	1.43	0.2069
Interaction	9.75	27	0.36111	0.72	0.8113
Error	20.00	40	0.50000		
Total	37.20	79			

SS, Sum of Squares; df: degree of freedom; MS, Mean Square.

4.5.3 Comparison with Related Works

We also compared our method with other methods that can be used to generate motion variations. An intuitive way to generate variations is to directly add noise to the original motion data. Here, we adopt the generally used Perlin noise [58] to generate variations. We visualized one DOF of the left foot joint for two motions



(a) (b)

Figure 4.7: One DOF of the foot joint across frames from two variants synthesized by adding Perlin noise. The red circle corresponds to the sawtooth peak of the curve. a) Normal walking motion; b) Fast walking motion.

generated by adding Perlin noise respectively, see Figure 4.7. The sawtooth peak of the curve is highlighted by the red circle in Figure 4.7. The sawtooth peak point is the moment that jerky movement happens. We can observe that the motions are not smooth as there are many sawtooth peak points of the curve shown in Figure 4.7. This is because variations in real human movement are not merely noise, but are constructing components of the motion itself. We also synthesized motion variations with the method proposed by Lau et al. [8]. The relations between joints are modeled with a second order Dynamic Bayesian Network. New motions can be synthesized by sampling from the predicted distribution with kernel regression. Snapshots of the variation results are shown in Figure 4.8. The variants from each methods are put in the same location on the ground for visualization. The blue characters represent the motions synthesized by our method. We can observe that motions from our approach show more variations, including the temporal and spatial variations. The motions synthesized by [8] showed less variations as they use the frame difference as the output feature while our approach use SCF as the output directly.

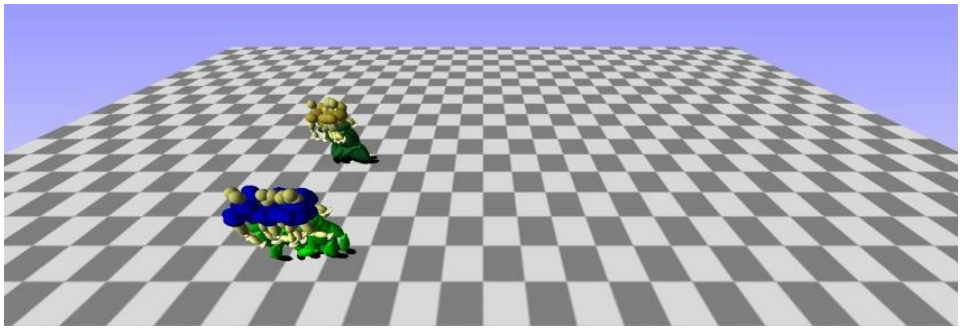


Figure 4.8: 10 variations of walking motion generate by our method and Lau et al. [8]. The variants from each method are put overlapped for visualization, respectively. The blue characters represent the results from our method and others are from Lau et al. [8]

4.6 Summary

In this chapter, we propose a novel generative probabilistic model to synthesize variations of human motion. Our focus differs from style transferring methods, which transfer the style of one motion to another. To be specific, we focus on variation generation within the same style of motion. Our method is appealing for human motion variation synthesis because our approach can learn from small training sets and the parameters of motion model can be optimized without relying on experimental cross-validation. The usage of SLFM can model the relation between the DOFs of each joint, which is more natural for human motion modeling. The skeleton representation is divided into multiple partitions to obtain the influence between joints. This partition based representation not only reduces the complexity of human motion but also helps us to define the influence between joints. The conditional dependency between joints is predefined for the joints within the same body partition based on the hierarchy structure.

Chapter 5

Human Posture Reconstruction

In Chapter 3, we proposed Temporal Sparse Representation (TSR) based human motion retrieval method for reusing motion capture data. It will be appealing if there are other alternatives that can capture human motions with easy to set up and cheap device. Recent advances in depth camera based motion tracking devices such as the Microsoft Kinect has enabled efficient human-computer interaction using body movement, enhancing interactive systems such as console games. Kinect is a controller-free hardware that infers 3D positions of human body joints from a single depth image with the help of motion recognition technology [37]. In this thesis, we RGBD camera (i.e. Kinect) as our purpose is for interactive applications and Kinect can estimate human poses in real time. However, RGB cameras alone are not suitable for interactive applications since it takes time for pose estimation from RGB cameras. While Kinect can be used to track the user and determine the user's 3D joint positions in a robust manner, which is convenient for posture control, the captured data suffers from poor precision due to self-occlusions. As illustrated in Figure 5.1, the blue skele-

ton represents the tracked result by Kinect SDK [88], in which some of the tracked joints are inaccurate when the upper body is occluded by the arms. The occlusion problem remains challenging despite the relevant research proposed in the past years. In this chapter, we propose a probabilistic model to reconstruct poses captured from Kinect. We adopt the Gaussian Process (GP) as a spatial prior to estimating the most likely correct pose given one posture data from Kinect. Unlike previous works that require the usage of a larger marker-based motion database, the GP based model can be robustly trained from small training sets. The parameters of the kernel function can be optimized without relying on experimental cross validation. We also introduce a temporal consistency term to constrain the velocity variations between successive frames. To ensure the reconstructed posture resembles the input pose from Kinect, we embed the reliability of each tracked joint into the posture reconstruction framework. The effectiveness of our approach is verified by reconstructing a number of motions containing self-occlusions. In the following of chapter, we will elaborate on the proposed framework for posture reconstruction with Kinect.

5.1 Data Acquisition and Preprocessing

For brevity, in this chapter we will use *MOCAP* to represent human motion data captured by an optical motion capture system. The postures obtained from Kinect are noisy and incomplete, whereas MOCAP is accurate and stable. Hence, we can use MOCAP to recover postures from Kinect.

In this chapter, we model the relationship between Kinect data and MOCAP

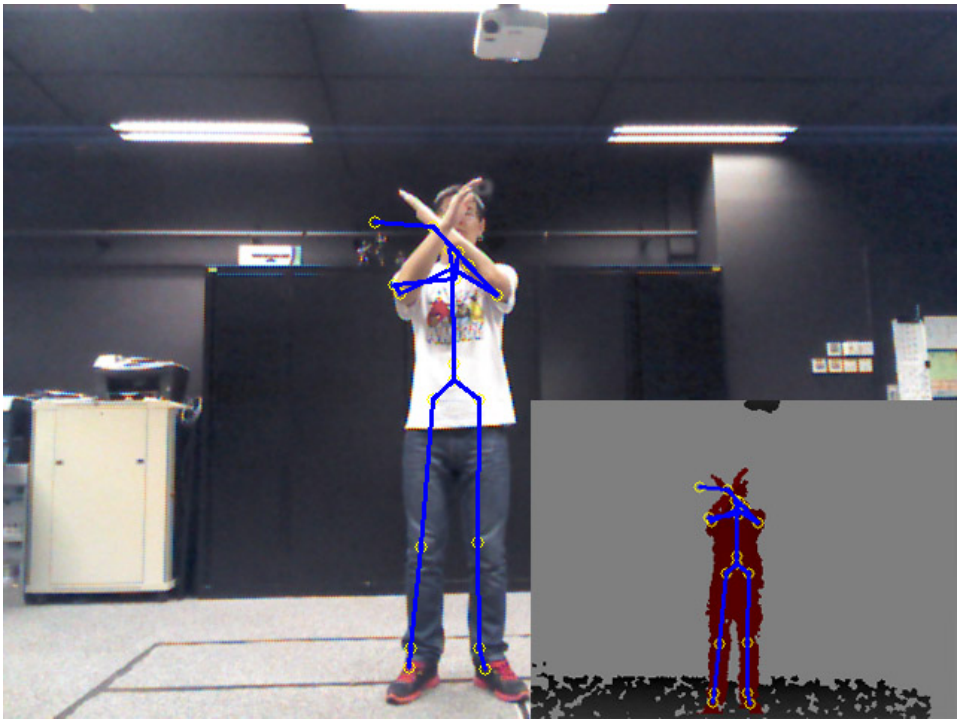


Figure 5.1: Example of an inaccurately tracked pose from Kinect.

with Gaussian Process. Specifically, we capture motions with Kinect and optical motion capture system at the same time to identify the correspondence between them. The setup of this capturing procedure is shown in Figure 5.2. The pose of Kinect at time t is denoted as $X_t = (x_t^1, x_t^2, ..., x_t^J), x_t^J \in R^3$, where x_t^J represents the 3D joint position of joint J over time t. There are 20 joints based on the skeleton definition of Kinect, i.e. $J = 20$. The corresponding MOCAP of X_t is denoted as $M_t = (m_t^1, m_t^2, ..., m_t^J), m_t^J \in R^3$.

To enhance the robustness of the spatial prediction model (Section 5.2.1) and to make the system invariant to different subjects, we normalize the postures for prediction. Specifically, the prediction model should output the same results given the same posture performed by different subjects when they are facing different directions. Here, we use the normalized data of X_t as the input of the predictor, that is $\tilde{X}_t =$

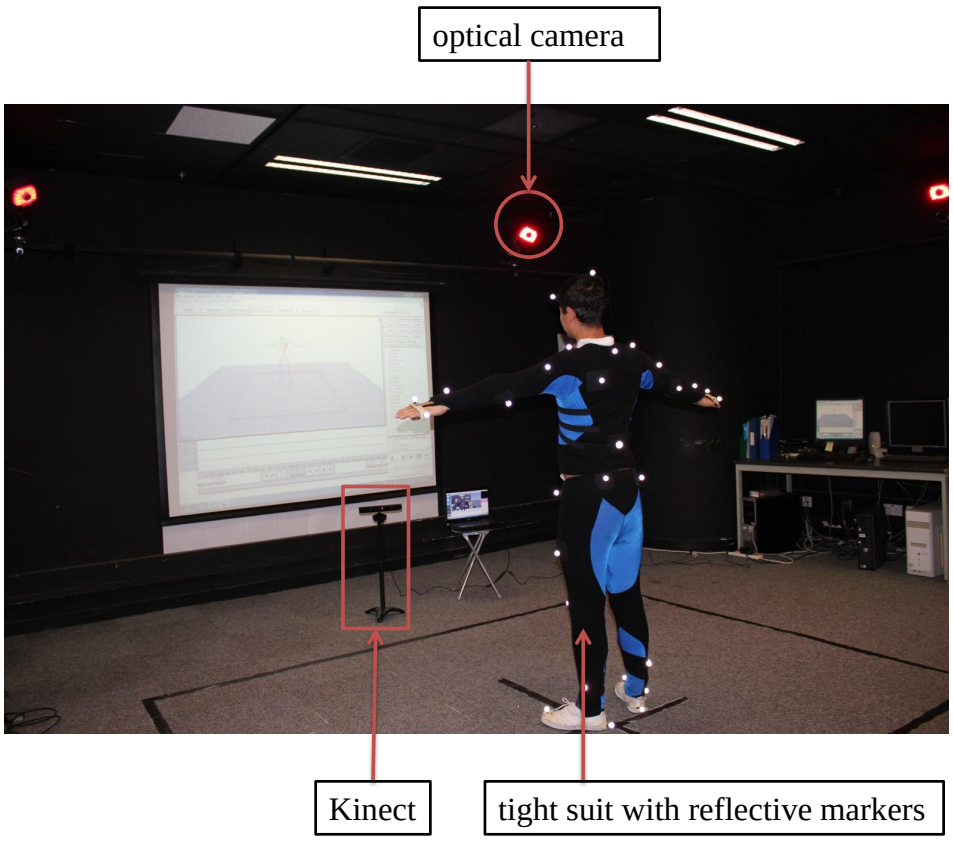


Figure 5.2: Human motion capture with Kinect and an optical motion capture system.

$F(X_t)$, where $F(\cdot)$ represents the posture normalization procedure and retargeting of the user's skeleton into a fixed skeleton size. The posture normalization procedure consists of two steps: removing the rotation along the vertical axis and the global 3-D translation. The retargeting procedure ensures the system to be invariant to the skeleton size of the user. We follow [85] to conduct the normalization and retargeting as it is simple yet effective.

5.2 Posture Reconstruction

To ensure that the reconstructed posture is accurate and resembles the input data from Kinect, we formulate the posture reconstruction as a optimization problem by minimizing an energy function. The energy function consists of a set of three energy terms to constrain the solution space, which are spatial prediction term, temporal prediction term, and reliability constraint. In the following, we will elaborate the definition and purpose of each term.

5.2.1 Spatial Prediction

Assuming that the MOCAP posture M_t is the corrected pose of the Kinect posture X_t, we design a spatial prediction term to evaluate how well the reconstructed posture fits with the MOCAP data, which implicitly favors solutions that are more similar to the correct posture.

Due to self-occlusions, there will be residual offset between X_t and M_t, which is calculated by $Y_t = M_t - X_t$, where $Y_t = (y_t^1, y_t^2, ..., y_t^J), y_t^J \in R^3$. During run-time, the objective is to predict the residual offset Y_t so that we can obtain the reconstructed pose M_t by appending Y_t to X_t. Obtaining Y_t is regarded as a prediction problem when given X_t. In this chapter, we adopt the non-parametric method Gaussian Process (GP) as the predictor. As GP based models can be robustly learned from small training sets, the usability of our algorithm is enhanced. GP is based on the assumption that adjacent observations should convey information about each other, and the coupling between observations takes place by means of the covariance matrix.

For more details about GP, we refer the readers to [104].

Human body joints are highly coordinated and it is important to take into account the relationship between them. Here, given one joint, we use its neighboring joints for prediction. Specifically, given joint J at time t, x_t^J, its neighboring joints $N(x_t^J)$ are defined as the set of joints that are directly connected with the same bone segment as joint J. Figure 5.3 shows three examples, the joints in green color are the neighboring joints of those in red color. For example, Figure 5.3(a) illustrates the neighboring joint of the right hand is right wrist. With such neighbor relationship extraction, the prediction model implicitly models the relationship between joints. Therefore, the input feature for obtaining y_t^i of joint i is the union set of $\tilde{x}_t^i$ and $N(\tilde{x}_t^J)$. $N(\tilde{x}_t^J)$ is the normalized data of the neighboring joints for joint J. Therefore, the input data is $\tilde{X}_t$ and $N(\tilde{X}_t)$, which correspond to $\mathbf{A}$. The output data is Y_t that corresponds to $\mathbf{B}$ of the prediction model. At training stage, with the obtained training data from Kinect and MOCAP, we can learn the hyper-parameters of the GP model.

With the learned model, we formulate the above prediction for Y_t as a conditional probability distribution, yielding the spatial prediction energy term as defined below:

$$\begin{aligned} E_S &= \ln p(Y_t|\tilde{X}_t, N(\tilde{X}_t)) \\ &= \frac{||Y_t - \mu(\tilde{X}_t, N(\tilde{X}_t))||^2}{2\sigma^2(\tilde{X}_t, N(\tilde{X}_t))} \end{aligned} \tag{5.1}$$

where μ and σ are the predicted mean and covariance functions from GP prediction respectively. The term E_S ensures that the reconstructed pose are close to the correct pose as much as possible. We use the residual offset as the output of the predictor because the system could explore more in the data space by predicting offset rather than

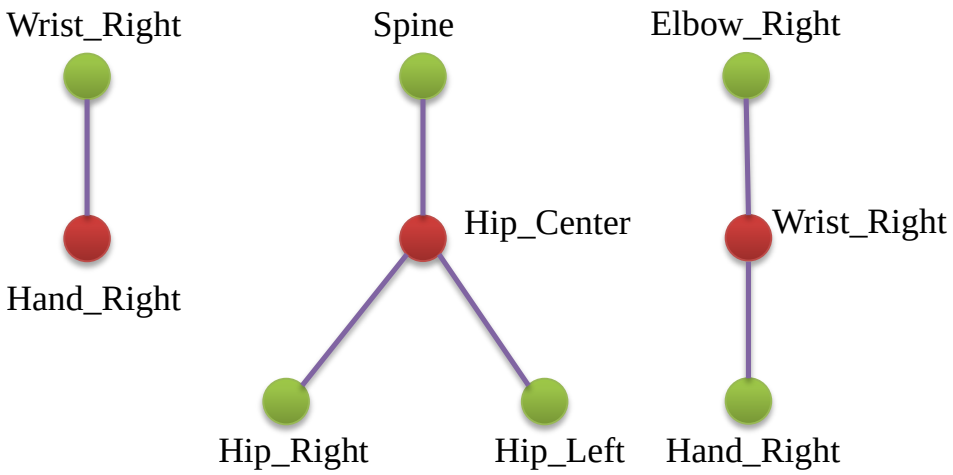


Figure 5.3: Three cases of neighboring joints. The red dot represents the joint for prediction, the green dots represent its neighboring joints. a) Right hand joint; b) Hip center joint; c) Right wrist joint.

the pose data directly, e.g. M_t. There are several publicly available implementations of Gaussian Process. In this chapter, we used the library developed by Lawrence [105]. To improve the performance of Gaussian Process prediction, we adopted the sparse approximation strategy proposed by Candela and Rasmussen [106].

5.2.2 Temporal Prediction

The above spatial prediction considers each posture independently. To ensure the temporal smoothness between consecutive frames, the relationship between frames is modeled as a second order temporal model, which has been verified to be effective to preserve temporal smoothness [107]. Specifically, we adopt a constant velocity variation to smooth velocity, which is formulated as below:

$$E_T = \ln p(M_t | M_{t-1}, M_{t-2}) \tag{5.2}$$

M_t, M_{t-1}, and M_{t-2} are the reconstructed postures at time slices t, $t-1$, and $t-2$. We have the following relationship between the reconstructed posture, input posture and the residual offset:

$$M_t = Y_t + X_t \tag{5.3}$$

Therefore, we can rewrite Equation 5.2 as :

$$\begin{aligned} E_T &= \ln p(Y_t + X_t | M_{t-1}, M_{t-2}) \\ &= ||(M_t - M_{t-1}) - (M_{t-1} - M_{t-2})||^2 \\ &= ||M_t - 2M_{t-1} + M_{t-2}||^2 \\ &= ||Y_t - (-X_t + 2M_{t-1} - M_{t-2})||^2 \end{aligned} \tag{5.4}$$

which facilitates the continuity in the reconstructed motions.

5.2.3 Reliability Embedding

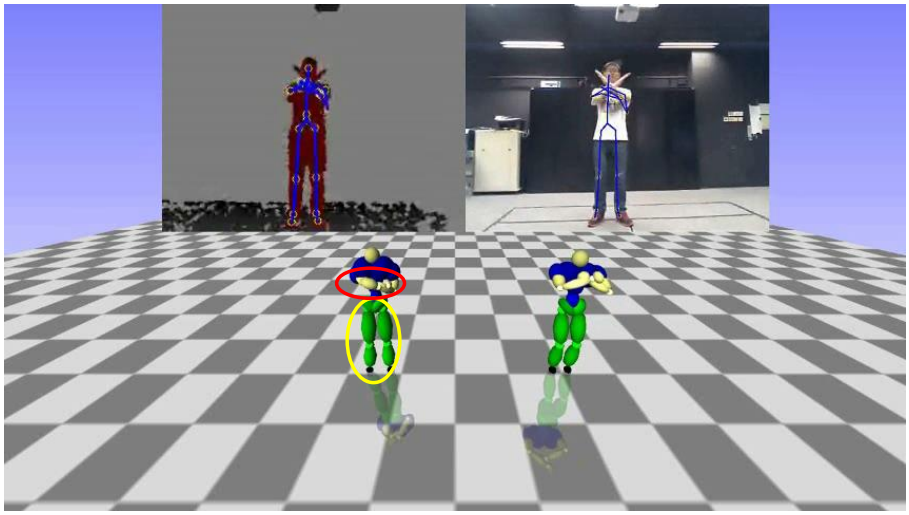
The accuracy of each tracked joint is different depending on the degree of occlusion. These incorrectly tracked joints from Kinect will incorrectly guide the system to infer the joint positions. The residual offset of these correctly tracked joints should be smaller as they are closer to the corrected posture, namely M_t. Thus, it is essential to consider the reliability of each joint to constrain the residual offsets of these joints with higher confidence during the prediction of Y_t. We use a reliability term E_R to penalize the residual offset of each joint based on its reliability, which implicitly ensures that the reconstructed pose resembles the input pose from Kinect as much as possible. More specifically, the residual offset value y_t^i of joint i should be smaller

if the corresponding joint is with higher reliability.

We adopt the strategy proposed by [85] to evaluate the reliability of the tracked joints from Kinect. They evaluate the reliability in three aspects: behavior reliability, kinematics reliability, and tracking state reliability. The behavior reliability refers to abnormal behavior of a tracked joint, which is calculated by the cosine similarity between two consecutive displacement vectors of one joint. The kinematics reliability represents the kinematic correctness of the tracked joints, which measures the change of bone length that connected with that joint. The tracking state reliability tells if a joint is tracked, inferred or not tracked when it is completely occluded. More details about the calculation of the reliability of each joint can be found in [85]. As a result, the reliability rate of each joint is a value between 0.0 and 1.0 (inclusive). We embed the reliability of each joint into the optimization framework and come up with the following reliability term:

$$E_R = ||RY_t||_F^2 \tag{5.5}$$

$||\cdot||_F$ is the Frobenius norm. The entry of R is the reliability of each joint, which ensures the reconstructed posture does not deviate from the input pose from Kinect. Intuitively, while minimizing the objective function, the value of y_t^i tends to be small when its reliability value is large. One example is shown in Figure 5.4, the joints in the yellow circle are with higher reliability, and our system tends to preserve these joints as much as possible. The joint joints in the red circle are with lower reliability, and our system tends to reconstruct these joints.



(a)

Figure 5.4: The left avatar is the result from Kinect and the right avatar is from our approach. The yellow circle represents joints with high reliability and the red circle represents joints with low reliability.

5.2.4 Energy Minimization Function

Equipped with the terms defined in the above sections, the posture reconstruction problem is formulated as the following optimization function:

$$E = \arg\min_{Y_t} \{w_S E_S + w_T E_T + w_R E_R\} \tag{5.6}$$

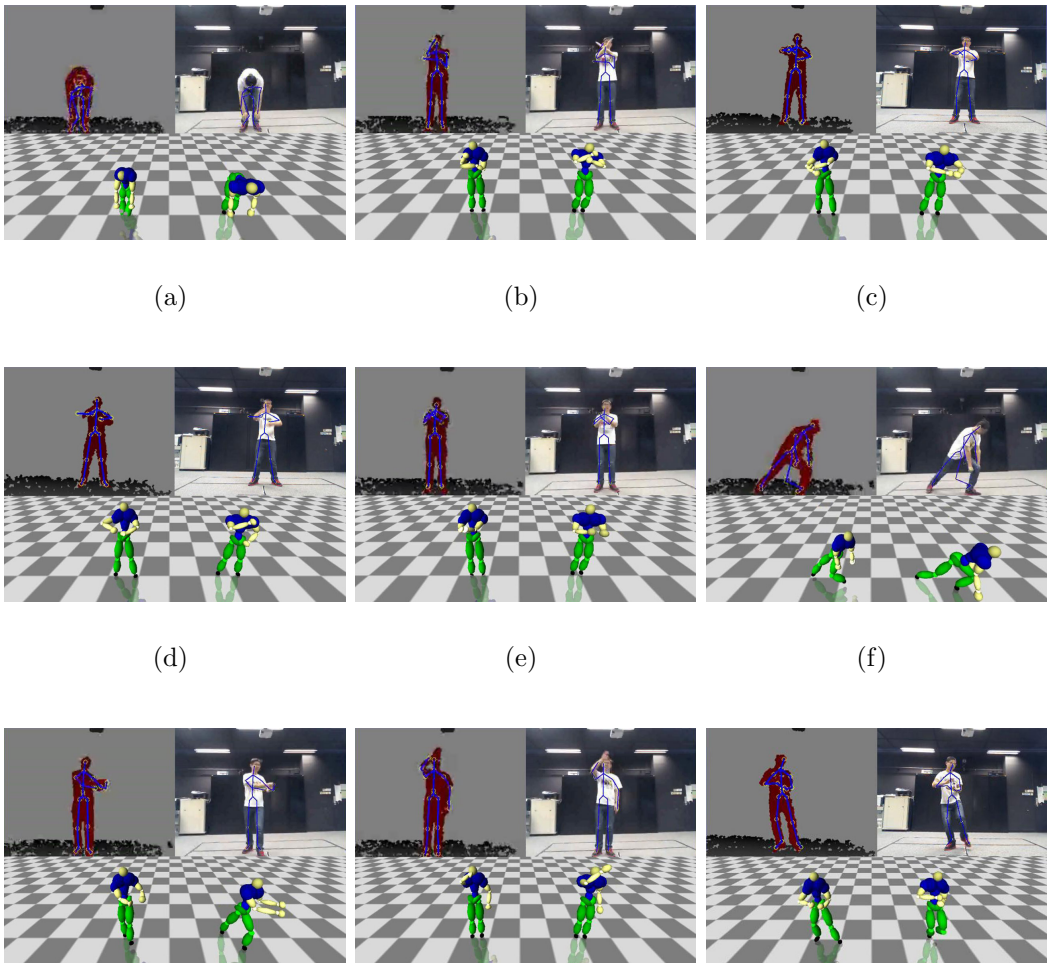
where w_S, w_T, and w_R are the weights of the energy terms. In our implementation, they are empirically set to be 0.6, 0.2, and 0.2, respectively. Our posture reconstruction system is a frame-based framework. The initial posture for optimization at each frame is defined as previous reconstructed posture, which makes the system has more chance to find the optimized posture. The optimization procedure stops when an optimal solution is found or the number of iterations reaches a predefined threshold.

There are some principles to tune the values of the weights. The weight of the spatial prediction term should be set the largest, since this term drags the reconstructed posture to the corrected pose as closely as possible. Second, the temporal prediction term ensures the temporal stability of the pose sequences. The details of the movements will be smoothed out as the term is based on a second order temporal prediction, which explicitly constrains the speed variation between consecutive frames. The reliability term makes sure the reconstructed posture is as similar as the Kinect posture, since the primary purpose of the system is to reconstruct Kinect postures. We will evaluate how these terms affect the accuracy of the system in Section 5.3.4.

In this section, we summarize the proposed method for posture reconstruction. At offline stage, we learn a spatial prediction model using Gaussian Process with pairwise Kinect data and marker-based motion capture data. It ensures the reconstructed posture as accurate as MOCAP data. At online stage, the system obtains an optimized posture with live captured data from Kinect, which ensures the reconstructed posture resembles the input pose from Kinect while maintaining the temporal smoothness between previous frames. Although we use pose data from Kinect, the overall framework is applicable to other system with different design of equations.

5.3 Experimental Results

In this section, we will show the experimental results and present the comparisons between alternative approaches such as the postures estimated by Kinect SDK [88]



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 5.5: Postures from Kinect and their corresponding reconstructed poses. The top two pictures are the depth and RGB image, in which the blue skeleton is the tracked results from Kinect. The left avatar in front represents the posture data from Kinect, and the right avatar corresponding to the postures reconstructed by our method. a) Bending over; b) Crossing arms; c) Rolling hands forward and backward; d) Rolling hands up and down; e) Clapping hands; f) Bending leg; g) Golf swinging; h) Waving right hand; i) Taichi motion.

and the work proposed by [87]. We first show results of posture reconstruction by our approach, where the postures are with severe self-occlusions. Qualitative and quantitative analysis are conducted to evaluate the accuracy between the proposed method and other alternatives.

The experimental results were conducted on a desktop computer with Intel Core 2 Duo 3.17 GHZ processor. The input data of our system are the postures estimated from Kinect. We obtain the 3D position of each joint through the latest official SDK (i.e. v1.8) provided by Microsoft [88]. An optical motion capture system with 7 cameras was used to capture MOCAP data in our own laboratory. The infrared emitters from the optical motion capture system will interfere with the Kinect. Here, we consider the Kinect device as one additional reflective marker of the optical motion capture system to eliminate the interference between Kinect and the optical motion capture system. The setup environment of Kinect and optical motion capture system is shown in Figure 5.2. On average, the system can reconstruct postures at 22 frame per second which is enough for realtime interactive applications.

5.3.1 Posture Reconstruction

The proposed approach works for users with different body sizes and proportions, because we use the normalized postures from Kinect as the input of the prediction model, where the rotation along the vertical axis and global 3D translation of each posture has been removed. We evaluate our system on a wide range of human motions, including sports activity such as Tai Chi, bending, golf swinging, and daily actions such as crossing arms, waving right hand, clapping hands, rolling hands up and down (rolling hands UD), rolling hands forward and backward (rolling hands FB). The size of each type of motion is reported in Table 5.1.

On average, the length of training data is around 15 seconds and around 450 frames for each type of motion, which is quite small compared with [85]. We choose

Table 5.1: The number of frames for each type of motion.

Motions	Tai Chi	Bending	Golf Swinging	Crossing Arms	Waving Right Hand	Clapping Hands	Rolling Hands UD	Rolling Hands FB	Bending Leg
Frames	650	320	460	380	350	420	480	475	385

these motions because all these motions contain severe self-occlusions, which are not well tracked by the Kinect system. However, the proposed method can well reconstruct these inaccurate poses even if a number of joints cannot be tracked by the Kinect sensor. For example, for the crossing hands motion, the joints of the body are occluded by the hands and the joints of the hands are occluded by each other. For a bending motion, the upper body joints are occluded by the arms. Figure 5.5 showed several frames of our results, the right avatar in front represents the postures reconstructed by our method and the left avatar corresponds to the estimated posture by Kinect SDK [88]. The top two pictures are the RGB image and depth image respectively. We can observe that certain parts of the postures from Kinect SDK are twisted when there exist occlusions whereas our method can reconstruct the postures very well. The readers are referred to the accompanying video for better visualization of the posture reconstruction results.

5.3.2 Qualitative Analysis

In this section, we evaluate the perceptual accuracy of the postures reconstructed by our method, postures from Kinect, postures by the method proposed by [87] and postures captured by an optical motion capture system. We use the Kinect SDK to obtain the 3D position of each joint as the Kinect data. [87] proposed to correct postures from Kinect with a random forest regression based approach without con-

sidering the reliability of each tracked joint from Kinect. The skeleton structure in the optical motion capture system and Kinect are different, and we need to carefully select these joints so that they are with the same skeleton structure definition. Here, we use the skeleton definition of Kinect as a reference skeleton structure. We measure the perceptual correctness for the postures of each method with a survey based evaluation [108], and it is also used in [85].

A total of 15 participants were invited to conduct this experiment. All of them had little or no experience about motion capture and 3D animation. The purpose of this experiment is to assess the relative correctness of the obtained postures from these four methods. We create a set of posture sequences with these four methods together with the RGB video so that the participants know what the actual actions are. Participants were asked to give a score for each motion based on its correctness without knowing the displayed motion come from which method. The score ranges from 1 to 10 (inclusive), where 1 means the most incorrect, and 10 means the most correct.

The score distribution for Kinect SDK [88], [87], our method and MOCAP is shown in Figure 5.6. The overall average scores of these four methods are 5.20, 6.42, 7.51, and 9.16 respectively, and the standard deviations are 0.84, 0.73, 0.71, and 0.57. It is expected that MOCAP data achieve the best scores. However, we can observe from the results that our method greatly outperforms Kinect and [87], which verifies our approach can improve the correctness of the postures output from Kinect sensor. We can see that for these motions with more occlusions, our method performs much better than Kinect and [87] such as bending and rolling hands. The reason is that we

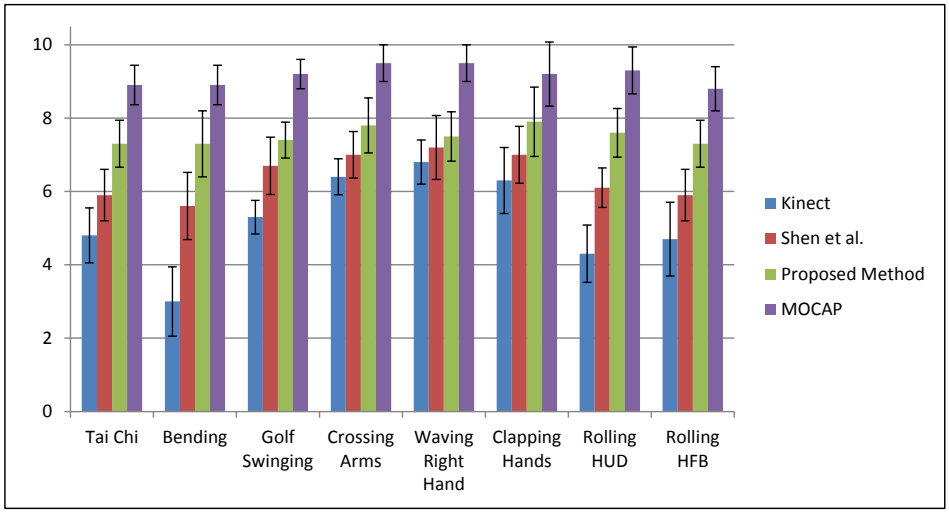


Figure 5.6: The score distribution based on the correctness of the postures from Kinect, Shen et al., proposed method and an optical motion capture system.

embed the reliability term into our optimization framework, which implicitly ensures the system tends to recover these joints more than other joints with higher reliability.

5.3.3 Quantitative Analysis

In this section, we quantitatively analyze the correctness of the proposed method. We can capture the postures with both Kinect and the optical motion capture system at the same time. We assume the data from optical motion capture system is the ground truth data. To evaluate the accuracy of the reconstructed postures, we define an error function to measure the distance between reconstructed postures and ground truth postures:

$$E(F_1, F_2) = \frac{1}{J} \sum_{j=1}^{J} E_j(F_1, F_2) \tag{5.7}$$

where F_1 and F_2 are the two sets of postures, J is the total number of joints. E_j is the reconstruction error of joint j between two set of postures, which is defined as:

$$E_j(F_1, F_2) = \sum_{t=1}^{T} D(F_{1t}^j, F_{2t}^j) \tag{5.8}$$

where T is the total number of postures and F_{1t}^j is the j-th joint of the posture at time t from F_1 . D is the Euclidean distance between two joints of two postures:

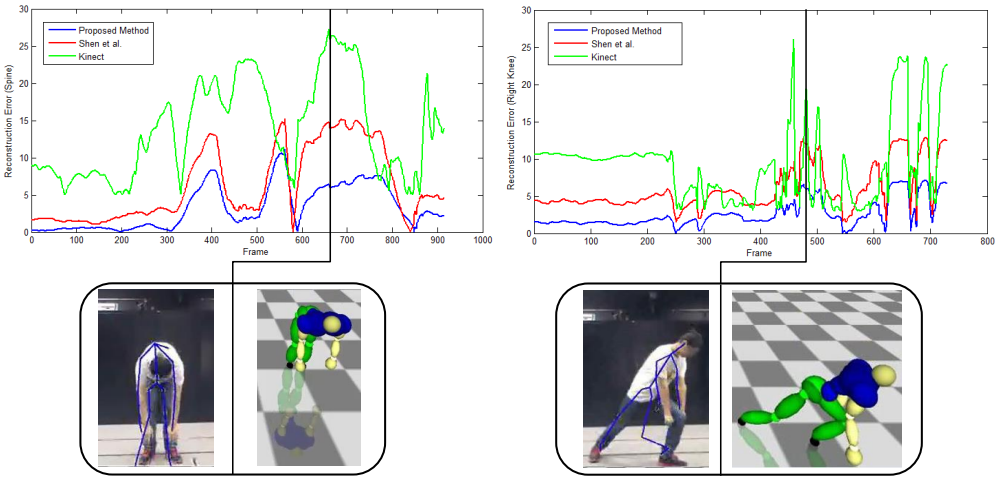
$$D(P_1^j, P_2^j) = \sqrt{(P_{1x}^j - P_{2x}^j)^2 + (P_{1y}^j - P_{2y}^j)^2 + (P_{1z}^j - P_{2z}^j)^2} \tag{5.9}$$

With the error function defined in Equation 5.7, we compare the Kinect raw data, postures reconstructed by [87], and postures reconstructed by our method with MOCAP data (unit: cm). Here we choose 5 types of motion for evaluation: clapping hands, crossing arms, bending, Tai Chi, and waving right hand. The results are shown in Table 5.2.

Table 5.2: Reconstruction errors of Kinect, Shen et al. and the proposed method.

Name	Number of Frames	Kinect (cm)	Shen et al. (cm)	Proposed Method (cm)
Crossing Arms	2052	12.5	9.8	7.2
Bending	1835	13.7	9.5	8.4
Tai Chi	2885	14.5	10.2	7.5
Waving Right Hand	1568	12.5	8.8	6.5

As expected, the errors from Kinect were large in general. Our method outperforms [87] as we take into account the reliability of each joint as the inaccurately tracked joint will guide the system to incorrectly infer the postures. For all classes of motions, our method consistently outperforms the other two, which verify the



(a) (b)

Figure 5.7: Examples of the reconstruction error of one joint across frames. The blue curve corresponding to our method, red curve is the reconstruction error of Shen et al. and the green curve is the reconstruction error of Kinect. a) Bending over motion; b) Bending leg motion.

effectiveness of the proposed method in terms of reconstruction accuracy.

To better observe the relationship between reconstruction error and reconstructed postures. We use the error function defined in Equation 5.9 for each joint and plot the reconstruction error across frames. Figure 5.7 shows the joint errors of two example motions. We can observe that both [87] and our method can achieve lower reconstruction errors compared with Kinect SDK. Another observation is that the trend of the error curves of [87] and our work tend to be similar, because both of these two works tried to reconstruct postures output from Kinect.

5.3.4 Performance Analysis

In this section, we analyze the reconstruction accuracy by examining the effectiveness of different terms in the objective function of Equation 5.6, respectively. The tempo-

ral prior term E_T ensures the temporal consistency between successive frames, which ensures the temporal smoothness between frames. The reliability term E_R encodes the reliability of each joint, which implicitly ensures the reconstructed posture resembles the Kinect observation as closely as possible. We computed the reconstruction error by dropping off the temporal term E_T and the reliability term E_R respectively. We used Tai Chi motion, bending over and crossing arms for evaluation because of their complicated movement features. The results are reported in Table 5.3. We

Table 5.3: Reconstruction error of the proposed framework with different constraint terms.

Setup	Terms Used	Reconstruction Error (cm)
(a)	E_T	15.8
(b)	E_R	13.4
(c)	E_S	11.7
(d)	E_R, E_T	12.8
(e)	E_S, E_T	10.2
(f)	E_S, E_R	9.4
(g)	E_S, E_T, E_R	7.7

found that both the temporal prior term and the reliability term improve the reconstruction accuracy, especially for the movements with severe self-occlusions. The result from setup (b) is better than setup (a), because E_T alone tends to generate motions with the minimum speed variations and the postures will deviate from the actual movements. Setup (c) is better than setup (a), setup (b) and setup (d) as use marker-based motion capture data as the spatial prior. It ensures the reconstructed posture close to mocap data much as possible. Although setup (f) achieves better results than setup (e), the obtained movements are jerky as setup (f) predicts postures independently without considering relationship between consecutive frames. The combination of these three terms can reconstruct postures with the least recon-

struction errors. Therefore, it is necessary to take into account the reliability of each joint and the temporal constraint during posture reconstruction.

5.4 Summary

In this chapter, we present a probabilistic framework to reconstruct live captured postures by Kinect. The Gaussian Process (GP) model is learned as a spatial prior to leverage position data obtained from Kinect and posture data from an optical motion capture system. GP is advantageous here, in that makes our system work even with small training data sets. We introduce a temporal consistency term into the optimization framework to minimize the discrepancy between current pose and previous poses. In addition, the reliability of each tracked joint is used to constrain the residual offset between pose from Kinect and true postures which ensures the reconstructed pose resembles the input pose from Kinect. The experimental results demonstrate that our system can achieve high quality poses even under severe self-occlusion situations and obtain higher accuracy compared with other alternatives.

Chapter 6

Conclusions and Future Directions

In this final chapter, the contributions made in the preceding chapters are summarized and some potential future research directions are presented.

The aim of all the works presented in this thesis is to reuse motion capture data by analyzing and synthesizing human motions, which may help animators to generate animations in a more efficient way and broaden interactive applications. An effective retrieval mechanism is essential for animators to search for a query motion from a large collection of motion capture database. In this thesis, we propose a framework for human motion retrieval based on a novel temporal sparse representation (TSR). Unlike previous methods, TSR takes into account the temporal information within each motion. To facilitate efficient comparison between TSRs, we propose a spatial temporal pyramid matching kernel in a coarse-to-fine manner, which has been verified to be effective in terms of retrieval accuracy. With the retrieved similar motions, we propose to reuse these similar motions for variation synthesis. Animators usually need to generate human motions manually for scenarios that require hundreds of

similar motions (i.e. crowd simulation). It is thereby appealing to develop a method that automatically synthesizes human motion variations. In this thesis, we develop a probabilistic framework for human motion variation synthesis based on a multivariate Gaussian Processes model, namely Semiparametric Latent Factor Model (SLFM). Our proposed method can synthesize human motions with more obvious variations compared other methods. In addition, it can learn from a small set of training data. Besides retrieving from motion capture database or synthesizing motions, another direction to obtain human motions for intended applications is to capture motions with low cost devices. In this thesis, we propose an optimization framework for posture reconstruction with Kinect. The proposed system generates postures for a wide variety of movements even for motions with severe self-occlusions, which is beneficial for real-time posture based applications such as motion-based gaming and sport training.

Here, we summarize the contributions of this thesis:

- **Human motion retrieval with temporal sparse representation**

  We propose a novel temporal sparse representation (TSR) for human motion retrieval. Compared with existing methods that adopt sparse representation, the proposed TSR encodes the temporal information within motions and thus generates a more compact and discriminative representation. In addition, a spatial temporal pyramid matching (STPM) kernel is designed based on T-SR, which can be used for logical comparison between motions. Moreover, our STPM kernel improves the effectiveness of motion retrieval in terms of retrieval

accuracy. Through our experimental evaluations, we demonstrate that our proposed human motion retrieval system has better performance and allows the user to retrieve desired motions from motion capture database. Finally, we implement a touch-less human motion retrieval system with Kinect. The proposed retrieval system allows the user to specify the query motion by performing it directly. Furthermore, the user interacts with the retrieval system using gestures so no controller is needed and the system delivers a more natural user interface. (Chapter 3)

◇ **Human motion variation synthesis**

We propose a novel generative probabilistic model to synthesize variations of human motion. The key idea is to model the conditional distribution of each joint via a multivariate Gaussian Process model, namely Semiparametric Latent Factor Model (SLFM). SLFM can effectively model the correlations between degrees of freedom (DOFs) of joints rather than dealing each DOF separately as implemented in existing methods. Detailed evaluations have been conducted to show that our proposed approach can effectively synthesize variations of different types of motions. Motions generated by our method showed a richer variations compared with existing methods. Finally, our user study show that the synthesized motions has similar level of naturalness as motion capture data. Our method therefore has great potentials to be applied in computer games and animations to enhance animation quality by introducing motion variations. (Chapter 4)

◇ **Human Posture Reconstruction**

We present a probabilistic framework to reconstruct live captured postures from Kinect. To overcome the incorrectly tracked and missing joints by Kinect, we adopt Gaussian Process (GP) model as a spatial prior to leverage position data obtained from Kinect and an optical motion capture system. Specifically, we model the residual offset between postures obtained from Kinect and mocap system instead of pairwise posture relationship. The residual offset modeling enables the system to cover more of the motion space. GP is advantageous here, in that makes our system work even with small training data sets. We introduce a temporal consistency term into the optimization framework by minimizing the discrepancy between current posture and previous postures. The reliability of the tracked joints from Kinect is different due to self-occlusions. To guide the optimized posture towards the input posture from Kinect, we incorporate the reliability of each tracked joint to constrain the residual offset between posture from Kinect and MOCAP data. Intuitively, the system tends to preserve the joint position value from Kinect as much as possible when the reliability of this joint is high. The experimental results demonstrate that our system can achieve high quality poses even under severe self-occlusion situations and obtain higher accuracy compared with other alternatives. Especially, the system can handle cases even when the motions involve a large number of occluded joints. (Chapter 5)

Although the works presented in this thesis have achieved fruitful results on

human motion analysis and synthesis, there are several possible directions for works presented in this thesis to be further studied.

◇ **Human motion retrieval**

For motion retrieval system that is based on temporal sparse representation, the parameters σ and l are determined experimentally and set to be fixed values for all types of motions. One future direction would be determining these parameters based on the motion types so that they can reflect the salient property of each particular motion type. The proposed temporal sparse representation (TSR) can be considered as a tensor representation, and tensor decomposition [109] will be another alternative for the matching between two TSRs. It is possible to scale up our method as the matching between motions is quite fast. However, it needs an index structure to store the motion data such as KD-tree. It will also be interesting to partition the body skeleton into spatial segments and combine with sparse representation as the importance of each partition varies between motions. For example, the upper body partition is more important for a boxing movement, whereas the lower partition is more important for a kicking movement. Hence, it will be interesting to propose a human motion retrieval system that allows the user to retrieve motions by partial matching based on the partition structure. In this thesis, we focus on single character motions. However, our framework can be extended for motions of multiple character. For single character motion, temporal sparse representation is calculated by the outer product of two frames within one motion. For

multiple character motion, the relationship between motions can be encoded by the outer product of two frames from two or multiple motions, in which the chosen frames are from different motions. Similar idea has been adopted by Tang et al. [110], where they used the inter distance between characters to capture the relationship between motions. The proposed retrieval system allows the user to input a query motion with Kinect. The relationship between the retrieval accuracy and the noise of the query motion is also one interesting direction to be worked on.

◇ **Human motion variation synthesis**

For human motion variation synthesis framework, it will be interesting to encode the contact information as a new feature into our model as described in Min et al. did [98]. SLFM allows different length scales for each input feature, which automatically determines the importance of each input feature to achieve feature selection. Meanwhile, generating variations in the semantic level is another possible future direction, which allows the user to control the degree of variation for the synthesized human motions. The proposed system needs a small set of training example motions. In the future, we will extend the proposed system to be a type-independent approach so that the framework can be used for different kinds of motions. Moreover, it will be interesting to extend the proposed framework for multiple characters. Chan et al. [28] proposed to synthesize two-character interactions by merging captured interaction samples with their spacetime relationships. It is possible to combine their approach

with our method so that we can synthesize variations for interactions between multiple characters.

◇ **Human posture reconstruction**

For the posture reconstruction framework, the assigned weights for the terms in the objective function are empirically set to be fixed in the proposed system. However, the weighs can be different for different types of motion to obtain optimal reconstructed postures. One possible solution would be to formulate the weight as a function of the residual offset, which is used to measure the importance of each term. Therefore, the weights can be adaptively determined according to the type of motion. The performance of our system will be affected by the speed of the movement since the tracking accuracy of Kinect will be affected by the speed. In the future, we will investigate the relationship between the temporal prediction term and the speed of the motions. The incorporation of physical constraints into the proposed framework is another interesting direction as the reconstructed postures in this work are not physically plausible. One possible way would be modeling the physical attributes (i.e. force field) between the Kinect data and mocap data as a prior distribution, and embed it into the optimization framework to generate physically valid postures. Capturing interactions between multiple characters is also one possible future direction. Liu et al. [111] proposed to capture motions of multiple characters using multi-view image segmentation, similarly, we can use multiple Kinects for posture reconstruction based on our method. Last but not least, integrating

our system with other simple yet stable devices such as motion sensor would be an interesting topic, because Kinect can only detect limited range of movements while motion sensor can be used as a complement, e.g. the user's back side.

Publications

Journal Papers:

1. **L. Zhou**, Z. Lu, H. Leung, and L. Shang, "Spatial temporal pyramid matching using temporal sparse representation for human motion retrieval," *The Visual Computer*, pp. 1-10, 2014.

   Best Paper Award of the 31st Computer Graphics International Conference, Sydney, Australia, 2014

2. **L. Zhou**, L. Shang, H. P. Shum, and H. Leung, "Human motion variation synthesis with multivariate gaussian processes," *Computer Animation and Virtual Worlds*, vol. 25, no. 3-4, pp. 303-311, 2014.

Conference Paper:

1. **L. Zhou** and H. Leung, "Human motion retrieval based on sparse coding and touchless interactions," in *Proceedings of the 2013 International Conference on Computer-Aided Design and Computer Graphics (CAD/Graphics)*, Nov 2013, pp. 417-418.

Bibliography

[1] M. Vondrak, L. Sigal, J. Hodgins, and O. Jenkins, "Video-based 3d motion capture through biped control," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 27:1–27:12, Jul. 2012.

[2] X. Wei, P. Zhang, and J. Chai, "Accurate realtime full-body motion capture using a single depth camera," *ACM Trans. Graph.*, vol. 31, no. 6, pp. 188:1–188:12, Nov. 2012.

[3] C. Wingrave, B. Williamson, P. D. Varcholik, J. Rose, A. Miller, E. Charbonneau, J. Bott, and J. LaViola, "The wiimote and beyond: Spatially convenient devices for 3d user interfaces," *Computer Graphics and Applications, IEEE*, vol. 30, no. 2, pp. 71–85, March 2010.

[4] A. Bleiweiss, D. Eshar, G. Kutliroff, A. Lerner, Y. Oshrat, and Y. Yanai, "Enhanced interactive gaming by blending full-body tracking and gesture animation," in *ACM SIGGRAPH ASIA 2010 Sketches*, ser. SA' 10, 2010, pp. 34:1–34:2.

[5] Z. Deng, Q. Gu, and Q. Li, "Perceptually consistent example-based human motion retrieval," in *Proceedings of the 2009 Symposium on Interactive 3D Graphics and Games*, ser. I3D '09, 2009, pp. 191–198.

[6] C. Sun, I. Junejo, and H. Foroosh, "Motion retrieval using low-rank subspace decomposition of motion volume," *Computer Graphics Forum*, vol. 30, no. 7, 2011.

[7] M. Zhu, H. Sun, and Z. Deng, "Quaternion space sparse decomposition for motion compression and retrieval," in *Proceedings of the 2012 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ser. SCA '12, 2012, pp. 183–192.

[8] M. Lau, Z. Bar-Joseph, and J. Kuffner, "Modeling spatial and temporal variation in motion data," *ACM Trans. Graph.*, vol. 28, no. 5, pp. 171:1–171:10, Dec. 2009.

[9] "The cmu motion capture library, http://mocap.cs.cmu.edu, 2005," 2005.

[10] M. Müller, T. Röder, M. Clausen, B. Eberhardt, B. Krüger, and A. Weber, "Documentation mocap database hdm05," Universität Bonn, Tech. Rep. CG-2007-2, June 2007.

[11] M. Müller and T. Röder, "Motion templates for automatic classification and retrieval of motion capture data," in *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ser. SCA '06, Vienna, Austria, Sep. 2006, pp. 137–146.

[12] M. Kapadia, I.-k. Chiang, T. Thomas, N. I. Badler, and J. T. Kider, Jr., "Efficient motion retrieval in large motion databases," in *Proceedings of the 2013 Symposium on Interactive 3D Graphics and Games*, ser. I3D '13, 2013, pp. 19–28.

[13] J. K. T. Tang, H. Leung, T. Komura, and H. P. H. Shum, "Emulating human perception of motion similarity," *Computer Animation and Virtual Worlds*, vol. 19, no. 3-4, 2008.

[14] Y. Li, C. Fermuller, Y. Aloimonos, and H. Ji, "Learning shift-invariant sparse representation of actions," in *Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition*, ser. CVPR '10, 2010, pp. 2630–2637.

[15] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition*, ser. CVPR '09, 2009, pp. 1794–1801.

[16] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *Image Processing, IEEE Transactions on*, vol. 15, no. 12, pp. 3736–3745, 2006.

[17] Z. Lu and H. H. Ip, "Spatial markov kernels for image categorization and annotation," *Trans. Sys. Man Cyber. Part B*, vol. 41, no. 4, pp. 976–989, Aug. 2011.

[18] N. Numaguchi, A. Nakazawa, T. Shiratori, and J. K. Hodgins, "A puppet interface for retrieval of motion capture data," in *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ser. SCA '11, 2011, pp. 157–166.

[19] T.-C. Feng, P. Gunawardane, J. Davis, and B. Jiang, "Motion capture data retrieval using an artistś doll," in *Proceedings of the 2008 International Conference on Pattern Recognition*, ser. ICPR '08, Dec 2008, pp. 1–4.

[20] M. G. Choi, K. Yang, T. Igarashi, J. Mitani, and J. Lee, "Retrieval and visualization of human motion data via stick figures," *Computer Graphics Forum*, vol. 31, no. 7pt1, pp. 2057–2065, Sep. 2012.

[21] M.-W. Chao, C.-H. Lin, J. Assa, and T.-Y. Lee, "Human motion retrieval from hand-drawn sketch," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 5, pp. 729–740, May 2012.

[22] H. Shum and E. S. Ho, "Real-time physical modelling of character movements with microsoft kinect," in *Proceedings of the 2012 ACM symposium on Virtual Reality Software and Technology*, ser. VRST '12, 2012, pp. 17–24.

[23] E. A. Suma, D. M. Krum, B. Lange, S. Koenig, A. Rizzo, and M. Bolas, "Adapting user interfaces for gestural interaction with the flexible action and articulated skeleton toolkit," *Computers & Graphics*, vol. 37, no. 3, pp. 193 – 201, 2013.

[24] G. Ren and E. O'Neill, "3d selection with freehand gesture," *Computers & Graphics*, vol. 37, no. 3, pp. 101 – 120, 2013.

[25] W.-S. Jang, W.-K. Lee, I.-K. Lee, and J. Lee, "Enriching a motion database by analogous combination of partial human motions," *The Visual Computer*, vol. 24, no. 4, pp. 271–280, Mar. 2008.

[26] Y. Ye and C. K. Liu, "Synthesis of responsive motion using a dynamic model," *Computer Graphics Forum*, vol. 29, no. 2, pp. 555–562, 2010.

[27] S. Jain and C. K. Liu, "Interactive synthesis of human-object interaction," in *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ser. SCA '09, 2009, pp. 47–53.

[28] J. C. Chan, J. K. Tang, and H. Leung, "Synthesizing two-character interactions by merging captured interaction samples with their spacetime relationships," *Computer Graphics Forum*, vol. 32, no. 7, 2013.

[29] W. Ma, S. Xia, J. K. Hodgins, X. Yang, C. Li, and Z. Wang, "Modeling style and variation in human motion," in *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ser. SCA '10, 2010, pp. 21–30.

[30] R. McDonnell, M. Larkin, S. Dobbyn, S. Collins, and C. O'Sullivan, "Clone attack! perception of crowd variety," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 26:1–26:8, Aug. 2008.

[31] M. Oshita, "Smart motion synthesis," *Computer Graphics Forum*, vol. 27, no. 7, 2008.

[32] E. Hsu, K. Pulli, and J. Popović, "Style translation for human motion," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 1082–1089, Jul. 2005.

[33] J. Min, H. Liu, and J. Chai, "Synthesis and editing of personalized stylistic human motion," in *Proceedings of the 2010 Symposium on Interactive 3D Graphics and Games*, ser. I3D '10, 2010, pp. 39–46.

[34] Y. W. Teh and M. Seeger, "Semiparametric latent factor models," in *Workshop on Artificial Intelligence and Statistics 10*, 2005.

[35] J. Chan, H. Leung, J. Tang, and T. Komura, "A virtual reality dance training system using motion capture technology," *Learning Technologies, IEEE Transactions on*, vol. 4, no. 2, pp. 187–195, April 2011.

[36] T. Schlömer, B. Poppinga, N. Henze, and S. Boll, "Gesture recognition with a wii controller," in *Proceedings of the 2008 International Conference on Tangible and Embedded Interaction*, ser. TEI '08, 2008, pp. 11–14.

[37] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, ser. CVPR '11, 2011, pp. 1297–1304.

[38] J. Chai and J. K. Hodgins, "Performance animation from low-dimensional control signals," in *ACM SIGGRAPH 2005 Papers*, ser. SIGGRAPH '05, 2005, pp. 686–696.

[39] H. Liu, X. Wei, J. Chai, I. Ha, and T. Rhee, "Realtime human motion control with a small number of inertial sensors," in *Proceedings of the 2011 Symposium on Interactive 3D Graphics and Games*, ser. I3D '11, 2011, pp. 133–140.

[40] G. Pradhan and B. Prabhakaran, "Indexing 3-d human motion repositories for content-based retrieval," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 13, no. 5, pp. 802–809, 2009.

[41] Y. Jin and B. Prabhakaran, "Semantic quantization of 3d human motion capture data through spatial-temporal feature extraction," in *Proceedings of the 2008 International Conference on Advances in Multimedia Modeling*, ser. MMM '08, 2008, pp. 318–328.

[42] K. Forbes and E. Fiume, "An efficient search algorithm for motion data using weighted pca," in *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ser. SCA '05, 2005, pp. 67–76.

[43] P. Wang, R. W. Lau, Z. Pan, J. Wang, and H. Song, "An eigen-based motion retrieval method for real-time animation," *Computers & Graphics*, vol. 38, pp. 255–267, 2014.

[44] F. Liu, Y. Zhuang, F. Wu, and Y. Pan, "3d motion retrieval with motion index tree," *Comput. Vis. Image Underst.*, vol. 92, no. 2-3, pp. 265–284, Nov. 2003.

[45] S. Tanuwijaya and Y. Ohno, "Tf-df indexing for mocap data segments in measuring relevance based on textual search queries." *The Visual Computer*, no. 6-8, pp. 1091–1100.

[46] T. Huang, H. Liu, and G. Ding, "Motion retrieval based on kinetic features in large motion database," in *Proceedings of the 2012 ACM International Conference on Multimodal Interaction*, ser. ICMI '12, 2012, pp. 209–216.

[47] S. Wu, S. Xia, Z. Wang, and C. Li, "Efficient motion data indexing and retrieval with local similarity measure of motion strings," *The Visual Computer*, vol. 25, no. 5-7, pp. 499–508, Apr. 2009.

[48] C.-Y. Chiu, S.-P. Chao, M.-Y. Wu, S.-N. Yang, and H.-C. Lin, "Content-based retrieval for human motion data," *J. Vis. Comun. Image Represent.*, vol. 15, no. 3, pp. 446–466, Sep. 2004.

[49] L. Kovar and M. Gleicher, "Automated extraction and parameterization of motions in large data sets," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 559–568, Aug. 2004.

[50] J. K. T. Tang and H. Leung, "Retrieval of logically relevant 3d human motions by adaptive feature selection with graded relevance feedback," *Pattern Recogn. Lett.*, vol. 33, no. 4, pp. 420–430, Mar. 2012.

[51] C. Chen, Y. Zhuang, F. Nie, Y. Yang, F. Wu, and J. Xiao, "Learning a 3d human pose distance metric from geometric pose descriptor," *IEEE Transactions*

on Visualization and Computer Graphics, vol. 17, no. 11, pp. 1676–1689, Nov. 2011.

[52] Y. Jin and B. Prabhakaran, "Knowledge discovery from 3d human motion streams through semantic dimensional reduction," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 7, no. 2, pp. 9:1–9:20, Mar. 2011.

[53] T. Qi, Y. Feng, J. Xiao, Y. Zhuang, X. Yang, and J. Zhang, "A semantic feature for human motion retrieval," *Computer Animation and Virtual Worlds*, vol. 24, no. 3-4, 2013.

[54] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, 2009.

[55] R. K. Ward and T. Guha, "Learning sparse representations for human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 8, pp. 1576–1588, 2012.

[56] R. Y. Q. Lai, P. C. Yuen, K. W. Lee, and J. H. Lai, "Interactive character posing by sparse coding," *CoRR*, vol. abs/1201.1409, 2012.

[57] Z. Tang, J. Xiao, Y. Feng, X. Yang, and J. Zhang, "Human motion retrieval based on freehand sketch," *Computer Animation and Virtual Worlds*, vol. 25, no. 3-4, 2014.

[58] K. Perlin, "Real time responsive animation with personality," *IEEE Transactions on Visualization and Computer Graphics*, vol. 1, no. 1, pp. 5–15, 1995.

[59] C. M. Harris and D. M. Wolpert, "Signal-dependent noise determines motor planning," *Nature*, vol. 394, no. 6695, Aug. 1998.

[60] C. Rose, B. Bodenheimer, and M. F. Cohen, "Verbs and adverbs: Multidimensional motion interpolation using radial basis functions," *IEEE Computer Graphics and Applications*, vol. 18, pp. 32–40, 1998.

[61] A. Safonova and J. K. Hodgins, "Construction and optimal search of interpolated motion graphs," *ACM Trans. Graph.*, vol. 26, no. 3, Jul. 2007.

[62] R. Urtasun, P. Glardon, R. Boulic, D. Thalmann, and P. Fua, "Style-based motion synthesis," *Computer Graphics Forum*, vol. 23, no. 4.

[63] J. Min, Y.-L. Chen, and J. Chai, "Interactive generation of human animation with deformable motion models," *ACM Trans. Graph.*, vol. 29, no. 1, pp. 9:1–9:12, Dec. 2009.

[64] Y. Li, T. Wang, and H.-Y. Shum, "Motion texture: a two-level statistical model for character motion synthesis," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 465–472, Jul. 2002.

[65] Y. Kim and M. Neff, "Component-based locomotion composition," in *Proceedings of the 2012 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ser. SCA '12, 2012, pp. 165–173.

[66] L. Ikemoto, O. Arikan, and D. Forsyth, "Generalizing motion edits with gaussian processes," *ACM Trans. Graph.*, vol. 28, no. 1, pp. 1:1–1:12, Feb. 2009.

[67] K. Grochow, S. L. Martin, A. Hertzmann, and Z. Popović, "Style-based inverse kinematics," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 522–531, Aug. 2004.

[68] N. D. Lawrence, "Gaussian process latent variable models for visualisation of high dimensional data," in *Proceedings of the 2004 Neural Information Processing Systems*, ser. NIPS '04, 2004, p. 2004.

[69] J. M. Wang, D. J. Fleet, and A. Hertzmann, "Gaussian process dynamical models," in *Proceedings of the 2006 Neural Information Processing Systems*, ser. NIPS '06. MIT Press, 2006, pp. 1441–1448.

[70] X. Wei, J. Min, and J. Chai, "Physically valid statistical models for human motion generation," *ACM Trans. Graph.*, vol. 30, no. 3, pp. 19:1–19:10, May 2011.

[71] M. Brand and A. Hertzmann, "Style machines," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, ser. SIGGRAPH '00, 2000, pp. 183–192.

[72] J. M. Wang, D. J. Fleet, and A. Hertzmann, "Multifactor gaussian process models for style-content separation," in *Proceedings of the 2007 International Conference on Machine Learning*, ser. ICML '07, 2007, pp. 975–982.

[73] I. Tashev, "Kinect development kit: A toolkit for gesture- and speech-based human-machine interaction," *Signal Processing Magazine, IEEE*, vol. 30, no. 5, pp. 129–131, Sept 2013.

[74] T. Morgan, D. Jarrell, and J. Vance, "Poster: Rapid development of natural user interaction using kinect sensors and vrpn," in *Proceedings of the 2014 IEEE Symposium on 3D User Interfaces*, ser. 3DUI '14, March 2014, pp. 163–164.

[75] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon, "Kinectfusion: Real-time 3d reconstruction and interaction using a moving depth camera," in *Proceedings of the 2011 Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '11, 2011, pp. 559–568.

[76] J. Han, L. Shao, D. Xu, and J. Shotton, "Enhanced computer vision with microsoft kinect sensor: A review," *Cybernetics, IEEE Transactions on*, vol. 43, no. 5, pp. 1318–1334, Oct 2013.

[77] S. W. Bailey and B. Bodenheimer, "A comparison of motion capture data recorded from a vicon system and a microsoft kinect sensor," in *Proceedings of the 2012 ACM Symposium on Applied Perception*, ser. SAP '12, 2012, pp. 121–121.

[78] H. Yasin, B. Krüger, and A. Weber, "Model based full body human motion reconstruction from video data," in *Proceedings of the 2013 International Conference on Computer Vision / Computer Graphics Collaboration Techniques and Applications*, ser. MIRAGE '13, 2013, pp. 1:1–1:8.

[79] N. K. Iason Oikonomidis and A. Argyros, "Efficient model-based 3d tracking of hand articulations using kinect," in *Proceedings of the 2011 British Machine*

Vision Conference, ser. BMVC '11. BMVA Press, 2011, pp. 101.1–101.11.

[80] G. Taylor, L. Sigal, D. Fleet, and G. Hinton, "Dynamical binary latent variable models for 3d human pose tracking," in *Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition*, ser. CVPR '10, June 2010, pp. 631–638.

[81] M. Vondrak, L. Sigal, and O. Jenkins, "Dynamical simulation priors for human motion tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 52–65, Jan 2013.

[82] J. Kim, Y. Seol, and J. Lee, "Human motion reconstruction from sparse 3d motion sensors using kernel cca-based regression," *Computer Animation and Virtual Worlds*, vol. 24, no. 6, 2013.

[83] T. Helten, M. Müller, H.-P. Seidel, and C. Theobalt, "Real-time body tracking with one depth camera and inertial sensors," in *Proceedings of the 2013 International Conference on Computer Vision*, ser. ICCV '13, 2013.

[84] M. Sigalas, M. Pateraki, I. Oikonomidis, and P. Trahanias, "Robust model-based 3d torso pose estimation in rgb-d sequences," in *Proceedings of the 2013 International Conference on Computer Vision Workshops*, ser. ICCVW '13, Dec 2013, pp. 315–322.

[85] H. P. H. Shum, E. S. L. Ho, Y. Jiang, and S. Takagi, "Real-time posture reconstruction for microsoft kinect," *IEEE Transactions on Cybernetics*, vol. 43, no. 5, pp. 1357–1369, 2013.

[86] 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 *Proceedings of the 2011 International Conference on Computer Vision*, ser. ICCV '11, 2011, pp. 1092–1099.

[87] W. Shen, K. Deng, X. Bai, T. Leyvand, B. Guo, and Z. Tu, "Exemplar-based human action pose correction and tagging," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition*, ser. CVPR '12, June 2012, pp. 1784–1791.

[88] M. Corporation, "Kinect for windows sdk beta programming guide version 1.8," 2013.

[89] G. Davis, S. Mallat, and M. Avellaneda, "Adaptive greedy approximations," *Constructive Approximation*, vol. 13, no. 1, pp. 57–98, 1997.

[90] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *Signal Processing, IEEE Transactions on*, vol. 41, no. 12, pp. 3397–3415, 1993.

[91] Y. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," in *Proceedings of the 1993 International Conference on Signals, Systems and Computers*, 1993, pp. 40–44 vol.1.

[92] S. S. Chen, D. L. Donoho, Michael, and A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, pp. 33–61, 1998.

[93] M. Aharon, M. Elad, and A. Bruckstein, "K-svd: An algorithm for designing overcomplete dictionaries for sparse representation," *Signal Processing, IEEE Transactions on*, vol. 54, no. 11, pp. 4311–4322, 2006.

[94] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proceedings of the 2006 IEEE Conference on Computer Vision and Pattern Recognition*, ser. CVPR '06, vol. 2, 2006, pp. 2169–2178.

[95] L. Mou, T. Huang, Y. Tian, M. Jiang, and W. Gao, "Content-based copy detection through multimodal feature representation and temporal pyramid matching," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 10, no. 1, pp. 5:1–5:20, Dec. 2013.

[96] F. S. Grassia, "Practical parameterization of rotations using the exponential map," *J. Graph. Tools*, vol. 3, no. 3, pp. 29–48, Mar. 1998.

[97] L. Deng, H. Leung, N. Gu, and Y. Yang, "Generalized model-based human motion recognition with body partition index maps," *Computer Graphics Forum*, vol. 31, no. 1, 2012.

[98] J. Min and J. Chai, "Motion graphs++: a compact generative model for semantic motion analysis and synthesis," *ACM Trans. Graph.*, vol. 31, no. 6, pp. 153:1–153:12, Nov. 2012.

[99] L. Shang and A. B. Chan, "On Approximate Inference for Generalized Gaussian Process Models," *ArXiv e-prints*, Nov. 2013.

[100] A. W. Feng, Y. Huang, M. Kallmann, and A. Shapiro, "An analysis of motion blending techniques," in *Proceedings of the 2012 International Conference on Motion in Games*, ser. MIG '12, 2012.

[101] L. Kovar, J. Schreiner, and M. Gleicher, "Footskate cleanup for motion capture editing," in *Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ser. SCA '02, 2002, pp. 97–104.

[102] E. Keogh and C. A. Ratanamahatana, "Exact indexing of dynamic time warping," *Knowl. Inf. Syst.*, vol. 7, no. 3, pp. 358–386, Mar. 2005.

[103] W. Penny and R. Henson, "Analysis of variance," in *Statistical Parametric Mapping: The analysis of functional brain images*, K. Friston, J. Ashburner, S. Kiebel, T. Nichols, and W. Penny, Eds. Elsevier, London, 2006.

[104] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

[105] "Gaussian process library, http://www.cs.manchester.ac.uk/neill/gp/," 2009.

[106] J. Quiñonero Candela and C. E. Rasmussen, "A unifying view of sparse approximate gaussian process regression," *J. Mach. Learn. Res.*, vol. 6, pp. 1939–1959, Dec. 2005.

[107] H. Sidenbladh and M. Black, "Learning image statistics for bayesian tracking," in *Proceedings of the 2001 International Conference on Computer Vision*, ser. ICCV '01, vol. 2, 2001, pp. 709–716 vol.2.

[108] L. Hoyet, R. McDonnell, and C. O'Sullivan, "Push it real: Perceiving causality in virtual interactions," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 90:1–90:9, Jul. 2012.

[109] Z. Wang and B. Vemuri, "An affine invariant tensor dissimilarity measure and its applications to tensor-valued image segmentation," in *Proceedings of the 2004 IEEE Conference on Computer Vision and Pattern Recognition*, ser. CVPR '04, vol. 1, June 2004, pp. I–228–I–233 Vol.1.

[110] J. K. Tang, J. C. Chan, H. Leung, and T. Komura, "Interaction retrieval by spacetime proximity graphs," *Computer Graphics Forum*, vol. 31, no. 2pt2, pp. 745–754, May 2012.

[111] Y. Liu, J. Gall, C. Stoll, Q. Dai, H.-P. Seidel, and C. Theobalt, "Markerless motion capture of multiple characters using multiview image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2720–2735, Nov 2013.

Appendices

Appendix A

TRC MOCAP Data Format

An example of *Motion Analysis TRC* (.trc) format of motion capture data:

Frame#	Time	Head_Top			LHead			RHead		
		X1	Y1	Z1	X2	Y2	Z2	X3	Y3	Z3
1	0	42.61	1722.16	2063.19	-36.59	1645.28	2154.8	116	1656.21	2159.52
2	0.017	42.59	1722.11	2063.39	-36.61	1645.33	2155.02	115.85	1656.18	2159.8
3	0.033	42.56	1722.08	2063.51	-36.68	1645.4	2154.95	115.59	1656.01	2160.04
4	0.05	42.45	1722.05	2063.64	-37.03	1645.35	2155.05	115.31	1655.95	2160.29
5	0.067	42.29	1722.04	2063.73	-37.25	1645.4	2154.87	114.97	1655.93	2160.52
6	0.083	42.08	1722.03	2063.76	-37.88	1645.23	2154.74	114.29	1656.06	2160.83
7	0.1	41.79	1722.02	2063.81	-38.28	1645.3	2154.77	113.89	1656.09	2160.99
8	0.117	41.44	1722.01	2063.8	-38.64	1645.23	2154.67	113.75	1656.05	2160.9
9	0.133	41.08	1721.95	2063.74	-39.02	1645.22	2154.63	113.37	1656.19	2160.91
10	0.15	40.63	1721.96	2063.67	-39.35	1645.21	2154.64	112.97	1656.33	2160.96
11	0.167	40.14	1721.92	2063.57	-39.77	1645.19	2154.57	112.56	1656.51	2160.96
12	0.183	39.75	1721.91	2063.52	-40.09	1645.18	2154.61	112.17	1656.66	2161
13	0.2	39.32	1721.88	2063.53	-40.41	1645.15	2154.67	111.81	1656.77	2161.09
14	0.217	39	1721.87	2063.53	-40.71	1645.11	2154.71	111.49	1656.86	2161.17
15	0.233	38.72	1721.83	2063.65	-40.99	1645.03	2154.8	111.18	1656.85	2161.34
16	0.25	38.49	1721.77	2063.83	-41.36	1644.99	2154.92	110.92	1656.84	2161.59
17	0.267	38.33	1721.75	2064.06	-41.64	1644.96	2155.06	110.44	1656.96	2162.03
18	0.283	38.19	1721.74	2064.43	-41.83	1644.9	2155.32	110.24	1656.89	2162.42
19	0.3	38.04	1721.67	2064.89	-42.02	1644.79	2155.65	110.02	1656.82	2162.86
20	0.317	37.87	1721.68	2065.43	-42.23	1644.63	2156.06	110.13	1656.64	2163.21
21	0.333	37.65	1721.7	2066.1	-42.37	1644.43	2156.56	109.97	1656.53	2163.81
22	0.35	37.54	1721.7	2066.75	-42.45	1644.25	2157.08	109.88	1656.39	2164.36
23	0.367	37.52	1721.66	2067.51	-42.44	1644.15	2157.81	109.81	1656.22	2165.03
24	0.383	37.55	1721.66	2068.37	-42.43	1643.97	2158.47	109.82	1656	2165.71
25	0.4	37.62	1721.65	2069.17	-42.35	1643.8	2159.14	109.9	1655.76	2166.34
26	0.417	37.72	1721.62	2070.01	-42.26	1643.62	2159.85	110.02	1655.52	2167.02
27	0.433	37.86	1721.6	2070.9	-42.18	1643.52	2160.66	110.19	1655.3	2167.71
28	0.45	37.99	1721.62	2071.75	-42.06	1643.37	2161.34	110.33	1655.08	2168.38
29	0.467	38.15	1721.61	2072.57	-41.94	1643.24	2162.05	110.19	1654.97	2169.2
30	0.483	38.37	1721.62	2073.38	-41.81	1643.17	2162.7	110.38	1654.75	2169.83
31	0.5	38.57	1721.64	2074.06	-41.67	1643.14	2163.29	110.56	1654.57	2170.48
32	0.517	38.72	1721.66	2074.76	-41.56	1643.13	2163.91	110.97	1654.32	2170.95
33	0.533	38.85	1721.72	2075.36	-41.4	1643.06	2164.44	111.09	1654.19	2171.43
34	0.55	38.94	1721.79	2075.96	-41.33	1643.03	2164.9	111.18	1654.1	2171.9
35	0.567	38.98	1721.83	2076.49	-41.27	1643.14	2165.56	111.22	1654.04	2172.33

Appendix B

BVH MOCAP Data Format

An example of *Biovision Hierarchy* (.bvh) format of motion capture data:

```
HIERARCHY
ROOT Hips
{
OFFSET 0 102.614 0
CHANNELS 6 Xposition Yposition Zposition Zrotation Xrotation Yrotation
JOINT RightUpLeg
{
OFFSET -8.94581 -8.66708 0.101652
CHANNELS 6 Xposition Yposition Zposition Zrotation Xrotation Yrotation
JOINT RightLeg
{
OFFSET 0 -44.4228 1.3727
CHANNELS 6 Xposition Yposition Zposition Zrotation Xrotation Yrotation
JOINT RightFoot
{
.
.
.
.
.
.
}
JOINT LeftToes
{
OFFSET -0.04 -4.41509 8.49911
CHANNELS 6 Xposition Yposition Zposition Zrotation Xrotation Yrotation
End Site
{
OFFSET 0.0800001 -0.0474481 9.7798
}
}
}
MOTION
Frames
Frame Time
```