ABSTRACT

Title of dissertation: COGNITIVE ROBOTS
FOR SOCIAL INTERACTION

Yi Li, PhD Student,
Electrical and Computer Engineering

Dissertation directed by: Professor Yiannis Aloimonos,
Computer Science

One of my goals is to work towards developing Cognitive Robots, especially with regard to improving the functionalities that facilitate the interaction with human beings and their surrounding objects. Any cognitive system designated for serving human beings must be capable of processing the social signals and eventually enable efficient prediction and planning of appropriate responses.

My main focus during my PhD study is to bridge the gap between the motoric space and the visual space. The discovery of the mirror neurons ([RC04]) shows that the visual perception of human motion (visual space) is directly associated to the motor control of the human body (motor space). This discovery poses a large number of challenges in different fields such as computer vision, robotics and neuroscience. One of the fundamental challenges is the understanding of the mapping between 2D visual space and 3D motoric control, and further developing building blocks (primitives) of human motion in the visual space as well as in the motor space.

First, I present my study on the visual-motoric mapping of human actions. This study aims at mapping human actions in 2D videos to 3D skeletal representation. Second, I present an automatic algorithm to decompose motion capture (MoCap) sequences into

synergies along with the times at which they are executed (or "activated") for each joint. Third, I proposed to use the Granger Causality as a tool to study the coordinated actions performed by at least two units. Recent scientific studies suggest that the above "action mirroring circuit" might be tuned to action coordination rather than single action mirroring. Fourth, I present the extraction of key poses in visual space. These key poses facilitate the further study of the "action mirroring circuit". I conclude the dissertation by describing the future of cognitive robotics study.

Cognitive Robots for Social Interaction

by

Yi Li

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2010

Advisory Committee

Professor Yiannis Aloimonos, Chair
Professor John Baras
Professor Min Wu
Doctor Cornelia Fermuller
Professor Gang Qu
Professor David Jacobs, Dean's representative

Acknowledgements

This dissertation would not have been possible without the advices from Professor Yiannis Aloimonos and Doctor Cornelia Fermuller. They helped me in all possible ways they can, and they are definitely among the greatest advisors I have ever seen.

I am also indebted to many of my labmates who supported me tremendously. They include Kostas Bitsakos, Justin Domke, Ji Hui, Ching Lik Teo, Yezhou Yang, Xiaodong Yu (in alphabetic order), and many others.

I also owe my deepest gratitude to my wife, Haizan Zeng, for her endless support and always being on my side.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

One of my goals is to work towards developing the functionalities that facilitate the interaction with human beings and their surrounding objects. Any cognitive system designated for serving human beings must be capable of processing the visual signals and eventually enable efficient prediction and planning of appropriate responses.

I have been working on a multi-disciplinary project that involves neuroscientists, robot engineers, and vision scientists. The project pinpoints the necessity of cognition modules that establish the protocols for parsing, generating and translating signals among body movements, visual objects and their linguistic descriptions. I strive to develop the multimodal tools for Social Signal Processing, including analyzing human actions in the visual space and the motoric space, and the coordination in human actions.

Previously, I worked on handwriting image analysis, and I developed a few practical tools that are reasonably recognized by the community. During this period, I realized the importance of a cognitive system in the applications where human body motion, such as hand movement in writing, is involved. Thus, I planned to go deeper in this direction, and I began to seek the research topics that are closely related to human's cognitive system and body motion.

1.1 Cognitive Functions for Human Interaction

My research on the mirroring mechanism for human-robot interaction analysis primarily aims at creating the sensorimotor representation of human action in visual space. Recently, application-specific action recognition has been widely studied. However, current computational approaches have been limited in their abilities to represent the unrestricted full-body motion and to learn the intrinsic features for semantic understanding of human action.

The study on Mirror Neurons provides the physiological mechanism for action perception, which states that one recognizes an action because he/she can represent the same ac-

tion using his own motoric representation. The implementation of such a cognitive system will shed light on the advancement of the human-centered research and the improvement of the quality of human life like humanoid, intelligent service robot, and diagnosis and rehabilitation of motion disorder diseases.

In the visual space, my goal is to develop a computational mechanism for mapping 2D movements in images to 3D body joint positions. A module for building a 3D skeleton from video data brings many advantages to computer vision. The skeleton is a natural abstraction of a human, which is viewpoint invariant and subject invariant. Besides being useful for human action analysis and recognition, it can be used in many applications in a number of related areas including computer graphics, visualization, and human-robot interaction with cognitive robots (*e.g.*, iCub [SMV07]).

In addition, I attempted to extract key poses for effective action recognition for robot-human interaction. It is well believed that not all poses are created equal, but much of the previous work largely focused on the recognition algorithms, and ignored to address the pose extraction sufficiently. I model the key poses as the discontinuities in the second order derivatives of the latent variables in the reduced visual space which we obtain using the Gaussian Process Dynamical Models (GPDM). Experiments demonstrate that the key poses are consistent under different conditions, such as subject variation, video quality, frame rate, and camera motion, and robust to viewpoint change. This facilitates the human action analysis and improves the action recognition rate significantly by reducing the uncharacteristic poses both in the training and the test sets. The results are also helpful in psychological experiments for action understanding.

In the motoric space, part of my work focuses on decomposing motion capture (MoCap) sequences into synergies (smooth and short basis functions) along with the times at which they are "activated" for each joint. The result will advance the humanoid research by providing effective building blocks for robot body movements. Given MoCap sequences, I proposed an algorithm to automatically learn the synergies as well as the activations simultaneously using L_1 Minimization, a novel optimization technique that aims at recovering the exact sparse solution by minimizing the L_1 norm (sum of absolute value) of the variables instead of the traditional sum of squared errors. Human actions by their nature are sparse both in action space domain and time domain. They are sparse in action space, because different actions share similar movements on some joints. They are sparse

in the time domain, because we do not want much overlap of the individual movements on a single joint. Two recent developed tools from machine learning, namely Orthogonal Matching Pursuit and the Split Bregman Algorithm, enabled us to alternately solve two large convex minimizations to learn the synergies and the activations simultaneously. Experiments demonstrate the power of the decomposition as a tool for representing and classifying human movements in the MoCap sequences. The activations of the synergies characterize the underlying rhythm in the parts of the bodies, and the weights of the synergies are effective for action retrieval and recognition. Particularly, the representation allows us to substantially compress novel MoCap data, which is fundamentally invariant to the sampling rate. Experiments further suggest that our approach is well suited for early diagnosis of motion disorder diseases (e.g., Parkinson's disease), an emerging issue in public health.

A direct application to the analysis of the motoric data is to study the human coordination. Among cognitive studies on human body movements, social coordination has gained more attention than others because human beings in any society have complex behaviors to coordinate themselves and interact with the real world for defensive, living and hunting purposes. In fact, recent scientific studies suggest that the "action mirror circuit", which maps the action in the visual space to the motoric space, might be tuned to social coordination rather than single action mirroring. Thus, it is very clear that the knowledge of social action coordination will be helpful for understanding the functionalities of the mirroring mechanism of human action, and will eventually result in efficiently estimating appropriate actions in response or in modifying behaviors online. In the project, we attempt to establish a theory for studying the influence in a group of coordinated actions using the concept of Mirror Neurons in neuroscience.

I proposed to use the Granger Causality as a tool to study the coordinated actions performed by at least two units. In the Granger Causality, actions are modeled by autoregressive processes, and the causality is framed in terms of predictability. If one action causes the other, then knowledge (history) of the first action should help predict future values of the latter. We successfully applied the Granger Causality to the kinematic data in a chamber orchestra to test the interaction among players and between conductors and players. As an extremely interesting case of the mirror-like mechanism, the motor systems in musicians support their orchestrated musical execution. We discovered that a good conductor

drives the orchestra more frequently than an amateur conductor, with a stronger driving force on average for all players in each piece of music. The above theory can be applied in other human interaction scenarios, such as sport game and dancing, to study the dynamical network of coordination.

1.2 Organization of the Dissertation

The remaining of this dissertation is organized as follows. Chap. 2 reviews the literatures in the related problems. Chap. 3 presents the learning approach for mapping 2D actions to 3D space. Chap. 4 presents the algorithm for learning motion primitives in MoCap data. Chap. 5 presents the study of coordinated actions. Chap. 6 presents the algorithm for automatic extraction of key poses in videos. Chap. 7 concludes the dissertation. My publications from 2004-2010 are listed in Appendix. A, and a brief resume is included in Appendix. B.

Chapter 2

Related work

2.1 Chapter Summary

This chapter reviews a number of areas related to human motion analysis. First, I give a brief introduction to the history of human motion capture. Second, I present the literature on human pose analysis in visual space. Based on the pose analysis, we can recognize human actions in videos. Third, I focus on analyzing human motion in the motoric space. In addition, I discuss the concept of action coordination and present a tool of modeling the coordination.

2.2 Brief History of Recording and Measuring Human Motion

Ancient philosophers studied animal motion for a long time, which dates back to 2000 years ago. The first known document in biomechanics written by Aristotle [Ari], tried to discuss the relation between the locomotion and the structure. But the computational study of human motion did not start until recently. 100 years ago when the "motion picture" was introduced, some pioneers began to explore the possibility of recording human motion using photography techniques. For example, Eadweard Muybridge setup a camera system to record different people performing different actions in motion [Muy].

The approach Muybridge used is now called "optical motion capture", which is a "non-invasive" approach. Usually multiple cameras are used in a room. Optical motion capture can be further categorized into "marker-based" and "markerless" approaches. In the marker-based approaches, multiple markers are sticked in different positions of the subject body, and are tracked across frames in realtime [VIC]. The markerless approaches usually use silhouettes [SC08]. Therefore, clean or static background and canvas are favored in the environment.

Unfortunately, due to the limitation of camera frame rates and image resolution, current state-of-the-art of computer vision systems cannot easily achieve high resolution in temporal and spatial domain at the same time. In addition, imperfect results of the image

segmentation techniques limit the usability and portability.

In the field of neuroscience, researchers develop different devices for measuring human motion. However, these devices are not portable for general purposes. One possible trade off between usability and accuracy is to use the inertial sensors. Recently, high accurate inertial sensors are used extensively in the field of computer graphics. Speed, gravity, and acceleration are usually taken into account. For example, MOVEN suit [MOV] is one of the mature products in the market. Depending on the applications, calibration might not be compulsory [YJSB09].

It is natural to use the skeleton as the model to control human motion. This model is usually called the stick-figure model, and it is used in different fields. For example, we can control a humanoid by specifying rotations of joint angles of a certain stick-figure model. In computer graphics, different animations are usually done by rendering different figure-stick models.

Joint rotation is frequently represented as Euler angle in MoCap datasets, which is the rotation angle with respect to the X, Y, and Z axis, respectively. One typical format, BioVision Hierarchy (BVH), stores the Euler angle directly in the plain text format. Since Euler angle has many fundamental flaws, quaternion [Qua] is widely used as an alternative in computer graphics community. It has many advantages compared to the Euler angle.

2.3 Human Pose Analysis

Human pose analysis in videos and MoCap sequences is very useful for compute vision and neuroscience studies.

2.3.1 Pose Analysis in Videos

2.3.1.1 Pose Estimation in Videos

Detecting humans in images has received significant attention in the area of computer vision. The studies on this topic can be categorized into: 1) pedestrian detection for tracking, 2) 2D pose estimation, and 3) 3D pose estimation.

Pedestrian detection is a very challenging task because of the variations in illumination, shadow, and pose. Many datasets have been collected (*e.g.*, [PSZ08], [DT05a]). Recently, Schwartz *et al.* [SKHD09a] used the Partial Least Squares (PLS) method to reduce the

dimension in the feature space of detection windows and outperformed other algorithms. Since the method was developed in the context of pedestrians, it is limited to full body human detection and walking actions.

Estimating arbitrary 2D poses from single images has been a challenging problem for a long time. Usually body parts detectors are used first to estimate the potential locations of the limbs, then skeleton models are imposed as constraints in finding the optimal locations of the limbs. To solve the 2D pose estimation problem, belief propagation and dynamical programming have been used. Ramanan [Ram06] parsed images to body parts using an iterative process to tune the performance. Based on Ramanan's result, Ferrari *et al.* [FMJZ] estimated a few upper body poses from the TV series "Buffy". Felzenszwalb and Huttenlocher [FH05] developed a part-based approach for object recognition and used it also for pose estimation. Recently, Sapp *et al.* [STT10] estimated body poses using cascade models. These approaches usually try numerous possible scales and rotations to find the optimal solution.

3D pose estimation directly from 2D images has also been of interest. Closely related to optical motion capture, multiple camera settings are very useful when the calibration data is available [SB06]. Algorithms for single uncalibrated cameras were proposed in [ARS09]. However, the explicit mapping functions were not well addressed in these studies. Urtasun and Darrell [UD08b] used Gaussian Processes as the regression technique to compute the mapping, but it is mainly single image based and the action recognition capability was not discussed.

2.3.1.2 Key Poses in Visual Space

Key pose has been used in action recognition [RA] and gesture recognition [SNI04] in visual space. Existing methods for pose extraction and temporal segmentation of action video can be categorized into two classes: 1) dynamics-based approaches and 2) model-based approaches. Dynamics based approaches, such as Marr's seminal work [MV82], try to detect the discontinuity between actions so that the boundaries can be detected. These approaches work well when background subtraction is applied [LN07]. Model-based approaches are tightly coupled with action recognition algorithms. The space-time approaches [KSH] and dynamic time warping [LJD09] techniques are frequently adopted.

Pose extraction in the motion capture (MoCap) sequences has also been studied. The

data from a motion capture suit are time series of three rotation angles each at a number of joints on the human body. Similar to the vision-based extraction, the methods can also be grouped to two categories. Jenkins and Mataric [JM02] use the KCS, a heuristic algorithm, to partition human motion using dynamics. Vecchio *et al.* [VMP03], Bissacco [Bis05] and Lu and Ferrier [LF04] assume that human motion is ruled by autoregressive (AR) processes or state-space models and partitioned the sequences based on different model parameters. In related work, neuroscientists use velocity, acceleration, and jerk as the measurements for input motion streams [Zat97].

A closely related topic to pose extraction is video temporal segmentation, which has been used for shot boundary detection, key frame extraction [TRE], video content analysis [XG08], and video synopsis [PRAP08]. Such segmentations are helpful for retrieving useful information from a large collection of videos. However, these techniques do not aim at extracting human poses. Action synopsis [ACCO05] is related to our work but the joint locations need to be labeled using a semi-automatic software called Icarus prior to the analysis.

Motion primitives can be conceived as smooth and nicely behaving functions between motion discontinuities. The discontinuities are characterized by the changes in acceleration of the muscle signals [ddSB03]. In model based approaches, the discontinuities in MoCap data are detected as the changes in the model parameters [Bis05].

In visual space, the motion discontinuities have often been used for key pose detection. Existing methods for pose extraction and temporal segmentation of action videos can be categorized into two classes: 1) dynamics-based approaches and 2) model-based approaches. Dynamics based approaches, such as Marr's seminal work [MV82], try to compute the discontinuities between actions in which the boundaries can be detected. These approaches are applicable when background subtraction is effective [LN07]. Model-based approaches are coupled to action recognition algorithms, and space-time approaches are frequently adopted [KSH].

2.3.2 Pose Analysis in MoCap data

A central problem in the analysis of MoCap data is how to decompose motion sequences into primitives. The representations of the primitives can be poses or basic atoms of actions over time.

2.3.2.1 Human Motion Primitives

Finding motion primitives has been studied in a large body of work. D'Avella *et al.* [ddSB03] discovered that the muscles were activated together to perform actions. [CJ06] applied non-negative matrix factorization to study torque patterns. [SMI07] studied motion using a dynamical system. None of these models uses shift-invariant primitives.

Li *et al.* [LFAJ10] recently proposed an unsupervised learning algorithm for automatically decomposing joint movements sequences into shift-invariant basis functions. These primitives assist the recognition, synthesis, and characterization of human actions.

A decomposition into shift-invariant features has been studied in acoustic signals classification ([BD06]). Shift invariant sparse coding [RKN07] further improved the performance of classification. A major difference between their mathematical formulation and ours is that we enforce the weights to be positive, and the basis functions of individual joints to be shifted coherently to realize an instance of an action.

L_1 minimization recently gained much attention with the emergence of compressive sensing [Can06] and has been applied frequently to image denoising [CJLS09], sparse representation of data [DGJL07], and for solving non-negative sparse-related problems [DT05b]. Our approach involves solving an L_1 norm minimization in many variables. Although in principle, it is possible to solve an L_1 minimization problem by formulating it as a linear programming problem, such an approach is not efficient when many variables are involved. But recently a number of fast algorithms have been developed for approximating the optimal solution. For example, Basis Pursuit [CDS01] solves the L_1 minimization by selecting the best bases. Orthogonal Matching Pursuit (OMP) [TG07] can reliably recover a signal with K nonzero entries given a reasonable number of random linear measurements of that signal. Alternatively, the Split Bregman Algorithm [GO09], approximates the optimal solution by iteratively solving efficiently a few simple sub-problems.

2.3.2.2 Temporal Segmentation of Human Motion

A few studies proposed methods for breaking MoCap sequences into small action segments. Jenkins and Mataric [JM02] used a heuristic algorithm to partition human motion. Vecchio *et al.* [VMP03], Bissacco [Bis05] and Lu and Ferrier [LF04] assumed that human motion is ruled by autoregressive (AR) processes or state-space models and partitioned the

sequences based on different model parameters. A comparison of partitioning algorithms in motor space can be found in [Bou08].

2.3.2.3 Applications

The action basis could find direct application in action embodiment [PHRA]. The segments are useful for action retrieval [DGL09] and action classification [YJSB09], and can be used for compressing human motion [LM] and [LFAJ10].

2.4 Action Recognition in Videos

There is a large body of work on recognizing actions in visual space ([BGS+05], [GKS02], [KWC], [SvG], and [SLC04]). 2D features from image sequences, such as optical flow, silhouette, and similarities between frames, are computed and mapped directly to semantic concepts such as walking [AT06], running [EBMM03] and dancing [RSH+05]. Lin *et al.* [LJD09] use poses, colors, and flow as features, and align a test video to possible training sequences.

There exist many algorithms for video abstraction (e.g., [PRAP08]). However, these techniques do not aim at analyzing human poses. Action synopsis [ACCO05] is related to our work but the joint locations need to be labeled using a semi-automatic software called Icarus prior to the analysis.

Action recognition can be categorized into three areas: 1) pose based methods, 2) trajectory based methods, and 3) interest points based methods. I will discuss related work in the first two categories in this chapter.

2.4.1 Pose Based Action Analysis

Lin *et al.* [LJD09] use the pose information, color, and the flow as the features, and align a test video to possible training sequences. Alternatively, 2D features from image sequences are mapped to the joint space. Urtasun and Darell [UD08a] recently use sparse probabilistic regression to inference the poses from the human silhouettes.

Dimension reduction techniques have been used to analyze human motion both in motor space ([CGM+], [WFH08]) and in visual space ([Ple03], [EL04], [WWW08]). Studies

show closed curves for periodic patterns in reduced spaces, but action recognition in the reduced signals has not been addressed.

2.4.2 Trajectory based Action Recognition

Action recognition has been intensively studied. We limit our discussion to the algorithms based on the trajectories of detected interest points. Using the Space-Time Interest Point [SLC04], Messing *et al.* [MPK] proposed an algorithm for computing the trajectories of tracked keypoints. Raptis and Soatto [RS] further improved the descriptor, and Matikainen *et al.* [MHS] outperformed other algorithms by considering pairwise relation in space. None of these approaches uses pose estimators or body joint locations, both of which are more representative than generic interest points.

2.5 Coordinated Action and its Causality Network

Causality is a well known concept used in reasoning, planning, and knowledge representation [MT97] in artificial intelligence [Pea00] and cognitive science [Ste08]. Given MoCap sequences, we can carry out scientific analysis in the related areas including cognitive science and neuroscience. In these areas, Causality is widely known for its essential support of reasoning, planning, and knowledge representation [MT97] in artificial intelligence [Pea00] and cognitive science [Ste08] because a change is possible only through an action of the causality [Kan87].

A change is possible only through an action of a causation [Kan87]. Thus, causality has many applications ranging from logic [GLL$^+$04] to humanoid robotics [Sca02]. A quantitative measurement of causality in correlated signals, such as coordinated human actions, will be very helpful for understanding the underlying connections between signals, and the messages passing through the connections.

Granger Causality was originally proposed by Granger to understand the influence between two correlated time series such as stock market indexes [Gra69]. Granger Causality is based on the modeling of forecasting. In the Granger Causality Test, linear predictors are used for evaluating the causality quantitatively.

This technique was recently adopted extensively in neuroscience to study brain activities [SE07]. Granger Causality is regarded as the flow of information from one part of the

brain to another, which helps determine whether a coincidence is the result of one process influencing another process [LDB+09].

Modeling of human motion using linear predictors is widely used in computer vision and machine learning [VMP03, LF04]. Linear predictors are used for approximating human motions in short time intervals sufficiently. Autoregressive (AR) processes are frequently used to model multi-dimensional trajectories of human body joints recorded by Motion Capture (MoCap) equipments.

Coordinated human action can be conceived as a result of the Mirror Neurons mechanism [FCO05], which maps actions in visual space to motoric space and leads to appropriate responses. Therefore, a causality network might be particularly useful for understanding the mechanism that facilitates anticipating other's actions and planning error corrections in motor control.

Coordinated human action in music performance recently attracted much attention because action prediction and error correction have already been shown to be a fundamental aspect of music performance [MRPK09]. Thus, Granger Causality may be ideal for developing models and techniques for measuring social interaction in a controlled framework [CVV09].

2.5.1 Autoregressive Model

In signal processing and time series analysis, an autoregressive (AR) model is a random process which is used to model and predict stochastic signals [Hay]. Related concepts include Moving average (MA) model, Autoregressive moving average (ARMA) model, and Linear predictive coding.

Calculation of the AR parameters requires computing the autocorrelation matrix of the process. This is done using the Yule-Walker equations or spectral methods.

2.5.2 Regression Analysis

Regression is extensively used in time series analysis and dimension reduction.

In general, regression refers to the methods that find the relations between observations and responses [DS98]. The goal of linear regression is to determine the values of the parameters for a linear function that best fit a set of observations. The regression is

called Multivariate Analysis [Cle94] (MVA) when the observations are multidimensional, and Multiple Regression (MR) when the responses are multidimensional.

In Multivariate Multiple Regression, the responses form a vector function. A common practice is to consider each response as an independent variable. Multivariate analysis then is applied to each dimension [DS98]. However, if the responses are correlated, latent variable methods and non linear methods must be used. Partial Least Squares [Abd07] was developed to find a linear regression model of the principal components in latent space and it considers the covariance in the output dimensions as well as the input dimensions.

Chapter 3

Learning Visual-Motoric Mappings of Actions

3.1 Chapter Summary

The visual descriptions of actions used in current vision algorithms are appearance-based, and thus viewpoint dependent. Humans, however, can recognize actions under varying viewpoints, because we use multimodal representations, combining information from the visual and motor space. Here we develop a 3D representation of actions based on motor and visual information. We learn the function that maps 2D image positions to 3D human body joint trajectories using synchronized videos in conjunction with Motion Capture (MoCap) data. Using this function, we then map video data from a single viewpoint to 3D motion descriptions, on which we perform action recognition.

As image data we use the positions of the hand, which are localized using color and motion in a Conditional Random Field (CRF) based segmentation, and the elbow locations, which are acquired using particle filtering. The function that maps 2D data to 3D is approximated using the Partial Least Squares (PLS) method. For every joint, the PLS model is trained to predict the 3D locations of the joint in future frames using its 2D motion history. We collected two dataset of 8 and 10 cooking actions repeatedly performed 8-10 times by 4 subjects, respectively. Synthetic and real experiments showed that our approach can robustly map 2D to 3D data, and the recognition accuracy on the 3D action data in comparison to the 2D data is improved by 17% and outperforms a state-of-the-art motion descriptor by 7.3% on our dataset.

This chapter is based on the paper submitted to the IEEE Conference on Computer Vision and Pattern Recognition 2011 and the paper submitted to the IEEE Conference on Robotics and Automation 2011. Please refer to [1, 2] in the Appendix A for details.

3.2 Introduction

3.2.1 Motivation

Humans have an amazing capability in visually recognizing other humans' actions [Joh73] despite the large variability in visual appearance due to differences in subject's style and viewpoint. Evidence from neurophysiological studies on the so-called "mirror neurons" points to mechanisms of multimodal representations of actions in humans. It is considered that the "action mirroring" [RC04] mechanism enables an agent to understand the actions of others in terms of his/her own motor representations, and thus this mechanism is at the core for the very basic yet essential capability of primates to perform imitation. Therefore, it necessitates a visual module that parses, maps, and reproduces human actions from videos to their own 3D skeletal representation.

State of the art vision algorithms focus on visual processing only, and use appearance-based approaches to represent actions. Clearly, these approaches have their limitations for generalizing to different viewing conditions. On the other hand, some studies analyze 3D data from motion capture suit, but do not consider vision. The challenging mapping problem from vision to action space has not been sufficiently addressed, and it is still unknown what action and vision representations exactly are associated with each other in primates. A few studies have discussed the association of visual data with action processes and proposed learning approaches to link the two modalities [MSN$^+$06, AGC10]. In this study we propose a computational mechanism for mapping 2D movements in images to 3D body joint positions. We express this mapping by a set of functions, which we approximate using linear methods (Fig.3.1).

A module for building a 3D skeleton from video data brings many advantages to computer vision. The skeleton is a natural abstraction of a human, which is viewpoint invariant and subject invariant. Besides being useful for human action analysis and recognition, it can be used in many applications in a number of related areas including computer graphics, visualization, and human-robot interaction with cognitive robots (*e.g.*, iCub [SMV07]).

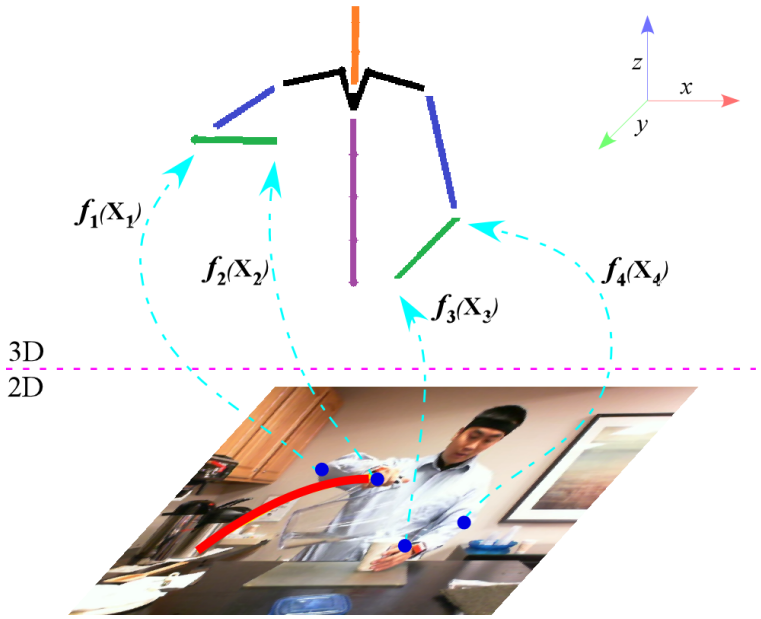


Figure 3.1: We aim to study the mapping $f_i(\cdot)$ from body joint history X_i ($i = 1..4$) in videos (red curve) to its 3D position (green: lower arm, blue: upper arm, purple: torso, orange: head).

3.2.2 Motion based Visual-Motoric Maps

We attempt to obtain the 3D movement of humans as this leads to invariance. Thus we are faced with the challenging problem of mapping 2D motion in images to 3D motion in space. Certainly, it is theoretically impossible to establish a one-to-one mapping between image points in a single view and their corresponding 3D points, because of the loss of the third dimension due to the projective transformation. Therefore, we will use the motion trajectories from video sequences to alleviate this problem.

We will map image points at joint locations, which we estimate using visual processes, to 3D joint locations obtained from MoCap data. The parameters of the mapping are computed from training data using regression. As the 3D representation is in a human-centered coordinate system, it is intrinsically viewpoint invariant. However, the learned 2D to 3D mapping depends on the viewpoint.

In order to use the proposed approach for general video interpretation, we will need to first determine the viewpoint from which the video is taken. Although we do not address this issue in this chapter, establishing the viewpoint from uncalibrated video may be done by detecting the orientation of human poses by the face and the shoulder alignment) with

respect to the camera. In our experiments, we used the frontal parallel viewpoint.

3.2.3 Our Approach in a Nutshell

We learn a mapping of the endpoints of human body limbs in 2D videos to 3D skeletons. The mapping is represented using a linear approximation. Learning of the parameters of the mapping is formulated as a time series problem, specifically, a multivariate multiple regression problem.

Our image representations are the 2D locations of the face, the elbows, and the hands. We first use a state of the art pose estimator to identify the hand locations and initialize a color model for the hand segmentation, which allows the approach to localize the hands in multiple frames in the videos. We also run an efficient face detector. Using the estimated locations of the face and the hands, a particle filter is then used to detect the elbows.

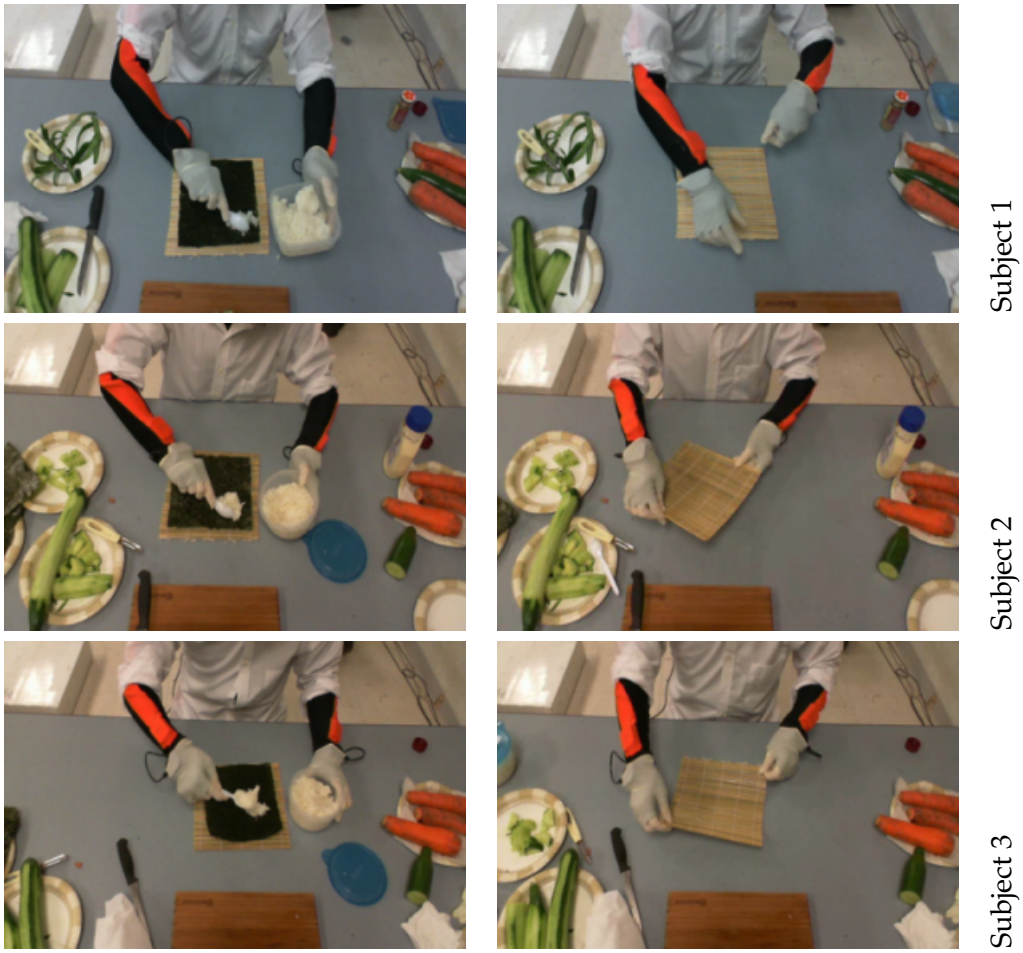
As 3D representation, we compute the 3D positions of the limbs, which are obtained from the MoCap data. Then we use the statistical linear method called Partial Least Squares (PLS) to compute the weights of the mapping.

3.2.3.1 The UMD-Sushi Dataset: A Pilot Dataset

As our experimental platform, we chose cooking actions. Our created a small UMD-Sushi dataset for a pilot study. It consists of eight different actions in sushi making, including "Cutting", "Peeling", "Pickup", "Pressing" "Placing", "Scooping", "Turning", and "Transferring". The actions were captured using the MOVEN motion suit at 100 frames per second (fps) and a high definition camera at 30 fps simultaneously.

The MoCap and video sequences were both resampled to 50 fps. Each action sequence consists of at least 10 repetitions of the same action. The MoCap coordinate system is the same as the conventional World Coordinate System (WCS), where the z coordinate increases vertically upward, and the coordinate system is left-handed.

Fig. 3.2 shows the same actions performed by different subjects in the visual domain. One can see there is reasonable intra-class variation for the same action type.



(a) Scooping (b) Turning

Figure 3.2: Two cooking actions in our UMD-Sushi dataset, Scooping (a) and Turning (b), performed by three different subjects in visual space.

3.2.3.2 Yet Another Cooking Dataset (YACD)

To estimate the mapping we need to detect accurately the body poses for a number of actions, and we need to have the MoCap data. The datasets currently available publicly (*e.g.*, [lTFHB$^+$, TBB]) do not satisfy these two requirements at the same time. Therefore, we captured our own pilot recording, which will be made available in public domain.

We recorded ten actions during cooking performed by four subjects. The actions are "Cleaning", "Cutting", "Flipping", "Peeling", "Picking", "Pressing", "Placing", "Sprinkling", "Stirring", and "Turning". The actions were captured using jointly the MOVEN motion suit [MOV], which was worn under natural clothes (Fig. 3.3), at 100 frames per second (fps) and a high definition camera at 30 fps. The videos feature half-body motions.

Similar to the UMD-Sushi dataset, The MoCap and video sequences were both resampled to 50 fps. Each action sequence consists of 8-10 repetitions of the same action. The MoCap coordinate system is the same as the right-handed World Coordinate System (WCS), with the z coordinate increasing vertically upward.

S_1 S_2 S_3 S_4

Figure 3.3: Examples of our dataset. The actions were captured using jointly the MOVEN motion suit [MOV], which was worn under natural clothes by the subjects S_1 to S_4, and a HD camera.

3.2.3.3 The Difference between Two Dataset

A significant difference between these two datasets is the viewpoint. The UMD-Sushi dataset primarily focus on the hand actions, and the colors of the motion capture suit can be easily detected. Therefore, it serves as a pilot dataset.

The YACD is a more formal dataset. Advanced computer vision algorithms are required to compute the human poses and to further localize hand positions. It is a more challenging dataset, thus, it is useful for demonstrating the effectiveness of the algorithm in real applications.

3.2.4 Our Contributions

1. We proposed a representation for the visual-motoric mapping of actions as a tool for recognition, using a linear approximation from image points to 3D poses.

2. Experiments on real and synthetic data demonstrate that our method improves action recognition against state of the art 2D visual representations by 7.3%.

3. We recorded an upper body cooking dataset using both MoCap suit and a HD camera, which complements current datasets.

The rest of the chapter is organized as follows. Sec. 3.3 presents the mapping algorithm.

Sec. 3.4 demonstrates the usefulness of our algorithm on synthetic and real data, and Sec. 3.5 concludes the chapter.

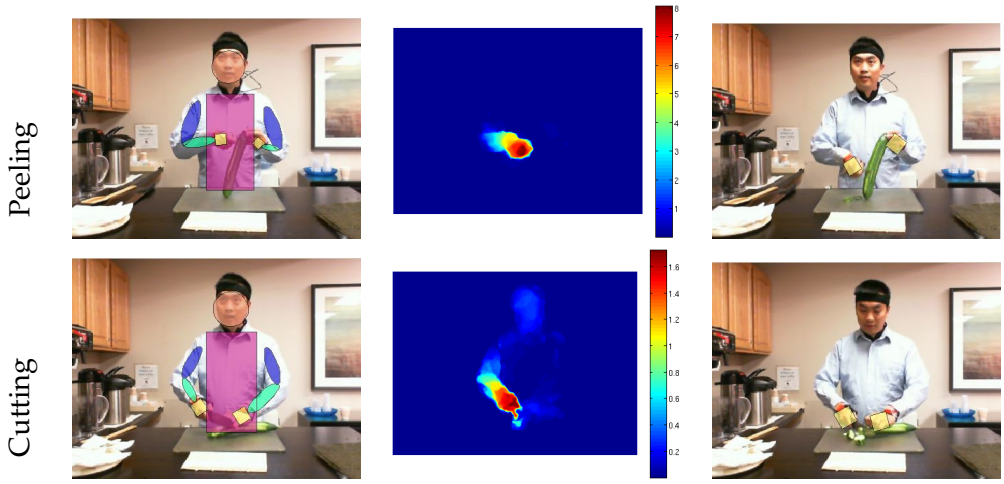


Figure 3.4: Our hand detector was initialized by a pose estimator (a). Then we used the trained color model and the optical flow (b) to localize the hands in the subsequent frames (c). Magnitude of the flow is color coded, and hand regions were coded in yellow.

3.3 The Visual-Motoric Mapping

To estimate the visual-motoric mapping, we extract body joint information in videos and compute 3D data of humans in the MoCap data, respectively. A MoCap sequence is the time series of three rotation angles each at a number of joints on the human body. In our approach, we used linear regression to approximate the mapping.

3.3.1 Accurate Localization of the Body Parts

Our goal is to accurately extract the body poses in video. Ideally, one can run a pose estimator on every frame. However, current pose estimators do not consider the smoothness of the poses over time. This is a limitation due to the quantization in scale and orientation used in estimating the pose. Furthermore, it is inefficient. To address these issues, we run a pose estimator [STT10] only once every 30 frames in videos, and detect the body joints in between. We describe the procedure in the following sections.

3.3.1.1 Hand Detector

Our hand detector is based on color and motion. The color model is automatically trained using the results of the pose estimator in [STT10] (Fig. 3.4a). For flow estimation, the implementation in [BM10] was adopted (Fig. 3.4b). Then, a Conditional Random Fields (CRF) model was adopted to localize the hand regions in the subsequent frames (Fig. 3.4c).

3.3.1.2 Brief Introduction to Conditional Random Fields (CRF)

We begin by providing the basic notations and background that are used to describe the CRF model in this chapter. Consider a discrete random variable $\mathbf{X}$ defined over the pixel lattice of the input image $I = \{1, \cdots, N\}$ with a neighborhood system $\mathcal{E}$. Each pixel, i in the lattice can then be viewed as a random variable $x_i \in \mathbf{X}$ that takes a value from the set of m possible labels $\mathcal{L} = \{l_1, \cdots, l_m\}$. $\mathcal{E}$ is a set of edges that connects the neighboring pixels or x_i within a clique, c. The set of all cliques is denoted as $\mathcal{C}$. We denote $\mathbf{x}_c$ as the set of random variables that are conditionally dependent on each other within each c; and $\mathbf{x}$ as a possible labeling (assignment) over I taken over the superset $\mathbf{L} = \mathcal{L}^N$. $\mathbf{L}$ therefore represents the space of all possible segmentations that can be applied on I. The goal of the CRF is to choose the labeling that satisfies certain constraints with highest probability. The probability of any labeling $\mathbf{x}$ is denoted by $P(\mathbf{x})$ and the constraints are modeled as potential functions ϕ. By the Hammersley-Clifford theorem, the posterior distribution $P(\mathbf{x}|\mathbf{I})$ where $\mathbf{I}$ is the set of observable data computed from I, is a Gibbs distribution:

$$P(\mathbf{x}|\mathbf{I}) = \frac{1}{Z} \exp\left(-\sum_{c \in \mathcal{C}} \phi(\mathbf{x}_c)\right) \tag{3.1}$$

where Z is the partitioning function, $\phi(\mathbf{x}_c)$ are the potentials defined over all x_i within the clique c, conditioned on data $\mathbf{I}$. Taking the negative log of (3.1) yields the Gibbs energy function:

$$E(\mathbf{x}) = -\log P(\mathbf{x}|\mathbf{I}) - \log Z = \sum_{c \in \mathcal{C}} \phi(\mathbf{x}_c) \tag{3.2}$$

Inference of the CRF involves minimizing Eq. 3.2 which gives us the maximum a posteriori (MAP) labeling, $\mathbf{x}^*$:

$$\mathbf{x}^* = \arg\min_{\mathbf{x}\in\mathbf{L}} E(\mathbf{x}) \tag{3.3}$$

and is the desired segmentation of image I. For the purpose of segmentation, Eq. 3.2 consists of potentials defined over unary and pairwise cliques:

$$E(\mathbf{x}) = \sum_{i\in I} \phi_i(x_i) + \sum_{(i,j)\in\mathcal{E}} \phi_{ij}(x_i,x_j) \tag{3.4}$$

The unary potentials, ϕ_i, encode the likelihood that a label, $\mathcal{L}$ is assigned to pixel i, which is computed from a learned distribution for the constraints that we are seeking. The pairwise potentials, ϕ_{ij}, encode consistency within segments and favor similarly labeled pixels within a clique and penalize the energy function when dissimilar labels appear within c.

3.3.1.3 Unary potential functions

We encode two unary potential functions into Eq. 3.4:

$$\phi_i(x_i) = \theta_{col}\phi_{col}(x_i) + \theta_{flow}\phi_{flow}(x_i) \tag{3.5}$$

where θ_{col} and θ_{flow} are the weighting parameters for the color, ϕ_{col}, and optical flow, ϕ_{flow}, respectively. ϕ_{col} is obtained from a Gaussian Mixture (GMM) color model, learned from training data over the CIELab color space. ϕ_{flow} is a unary prior that encodes the motion of hand and tool. Using it for segmentation favors moving regions. ϕ_{flow} is obtained from a bimodal GMM of optical flow learned from the training data. We use the implementation by Brox and Malik [BM10] to compute the flow. By combining the color and flow potentials, we induce a strong prior on hand-like regions that are moving.

The weighting parameters θ_{col} and θ_{flow} can be further adjusted to favor the final segmentation either towards the color model (to get more hand-like regions) or the flow model (to get regions with higher flow). To obtain a segmentation that favors the hand color model, we set $\theta_{col} > \theta_{flow}$ such that the posterior likelihood computed from the hand color GMM is greatly increased. This means that pixels with color closer to the hand color will be labeled as hand, even if the flow in that region is small. Conversely, in order to obtain a segmentation that favors regions of high flow, we set $\theta_{flow} > \theta_{col}$ to favor the flow

model.

3.3.1.4 Pairwise potential functions

In order to enforce consistent labels within segments, the pairwise potentials are defined as:

$$\phi_{ij}(x_i, x_j) = \begin{cases} 0 & \text{if } x_i = x_j \\ f(x_i, x_j) & \text{otherwise} \end{cases} \tag{3.6}$$

where $f(x_i, x_j)$ is an edge based contrast function defined over the image gradient, color and flow difference:

$$f(x_i, x_j) = \pi \exp(-\beta \left\| x_i - x_j \right\|^2) \tag{3.7}$$

where $\beta = (2 * \left\langle (x_i - x_j)^2 \right\rangle)^{-1}$ and $\langle \cdot \rangle$ represents taking the mean. π is a constant parameter that is different for the gradient, color and flow features. Eq. 3.7 favors color and flow constancy within regions of similar color or flow by penalizing less the potentials when similar labels are assigned within a clique. This formulation had been widely used in several state of the art color segmentation algorithms ([BJ01, RKB04]) with impressive results.

First we have to obtain the potentials in $E(\mathbf{x})$. Then given a test image, we apply Eq. 3.3 to compute the hand and flow segments, which are used to localize the tools in the hands. The following sections elaborate these processes in detail.

3.3.1.5 Segmenting Hands and Flow

The system must initially be trained to obtain the GMM color and flow models. This is done by either randomly sampling 5 frames from the training sequence (each of length 100 frames) where binary labels of the color classes (hand and non-hand regions) and flow classes (large flow and small flow) are manually assigned (Fig. 3.5), or use a pose estimator to initialize the hand regions.

This training process is equivalent to the service robot being shown typical examples of hands and flow training data during a short period of teaching the robot. Since the

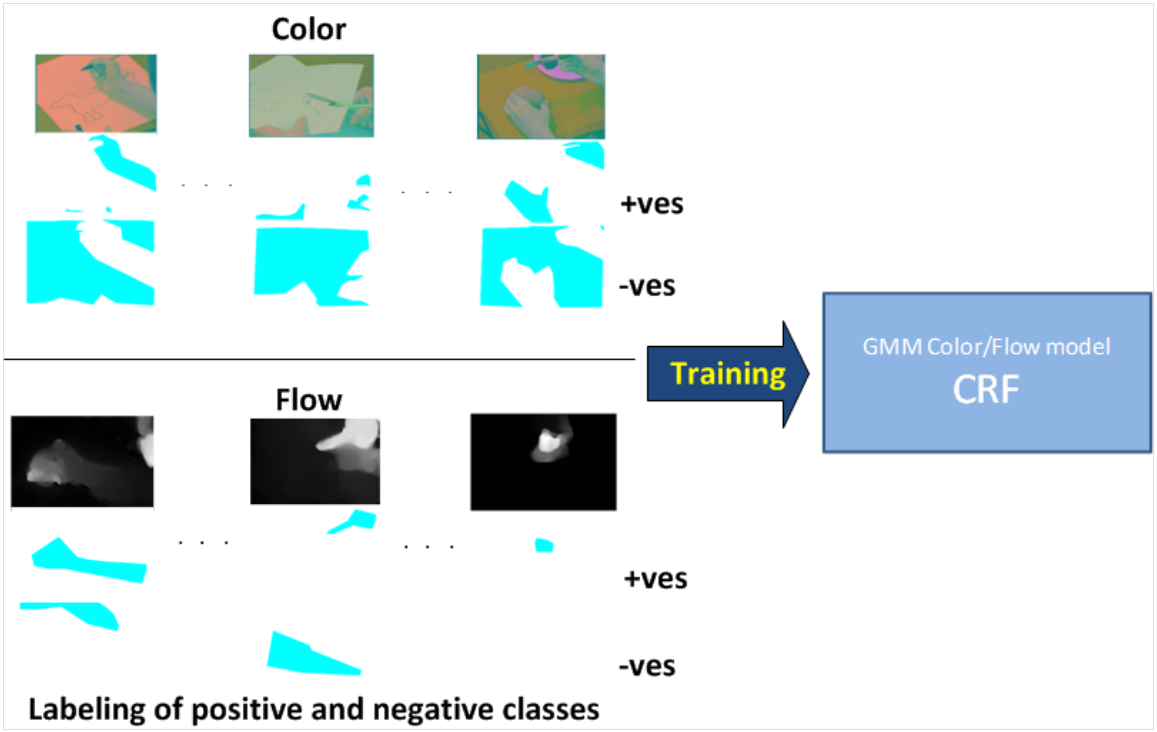


Figure 3.5: Training the GMM color and flow model. "+ves" and "-ves" denote the manually labeled positive and negative regions for the color (top) and flow (bottom) images. Figure courtesy of Ching Lik Teo.

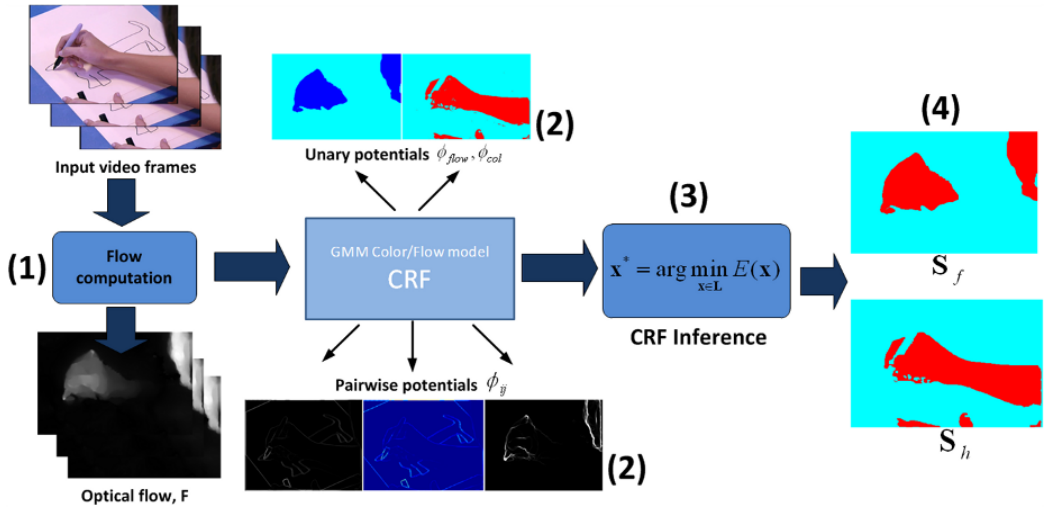


Figure 3.6: Segmenting hands and flow from a test image sequence. Labels with numbers correspond to the description. (1) Compute the optical flow. (2) Compute the unary and pairwise potentials. (3) Perform inference to get the best labels to obtain the hand and flow segmentation in (4). Figure courtesy of Ching Lik Teo.

class labels are typically invariant across many testing scenes (e.g. hand color and hand actions are unlikely to change drastically), it is expected that training will only be necessary during the initial stages of the service robot's life or when large changes to its environment or actions occur (e.g. from typical hand movements to large body movements).

During the testing phase, a sequence of test image frames are presented to the algorithm which computes $\mathbf{S}_h$ and $\mathbf{S}_f$ in the following steps as illustrated in Fig. 3.6:

1. Compute the optical flow $\mathbf{F}$ using the implementation of [BM10] between the previous and current image frames.

2. Compute the unary and pairwise potentials from Eqs. 3.5 and 3.6.

3. Obtain two energy functions $E(\mathbf{x})$ for the hands and flow by adjusting the unary potentials weights θ_{col} and θ_{flow} in Eq. 3.4. This is done by setting $\theta_{col} > \theta_{flow}$ to favor the hand regions in the first case and $\theta_{flow} > \theta_{col}$ to favor the high flow regions in the second case. Since we still want consistent labels within the segments, the pairwise potentials ϕ_{ij} are the same in both cases.

4. Perform α-expansion optimization on both energy functions using Eq. 3.3 to obtain $\mathbf{S}_h$ and $\mathbf{S}_f$.

3.3.1.6 Elbow Tracker

We used particle filtering to estimate the elbow positions, based on the hand and face detection results.

The Viola-Jones face detector was first used to estimate the faces. We used the offset between the face and the shoulder given by the pose estimator in Sec. 3.3.1.1 to compute the shoulder positions, and then together with hand positions we used a particle filter on each limb to estimate the positions of the elbows over time.

In particle filtering, we randomly generate 300 hypothesis for the pose positions, and keep $k = 10$ in each stage. For each guess, we measure the similarity by the difference in color distribution between the current and the previous regions of the upper arm and the lower arm, respectively.

Fig. 3.7 shows the results for our 2D pose localization on four different actions performed by four subjects. The torso was detected using Calvin body detector[1]. Poses in consecutive frames were correctly estimated. Please refer to Fig. 3.1 and 3.4 for the coding of the colors.

3.3.2 Learning the Mapping between 2D video and 3D MoCap Trajectories

Without loss of generality, we assume that the videos and the MoCap data were synchronized. Denote the locations of the two endpoints of a body part in 2D over time at n time instances as:

$$s = [p_1, p_2, ..., p_{\mathrm{t}}, ..., p_{\mathrm{n}}] \tag{3.8}$$

and the corresponding points trajectories in 3D as:

$$S = [P_1, P_2, ..., P_{\mathrm{t}}, ..., P_{\mathrm{n}}] \tag{3.9}$$

where p_i and P_i are column vectors. Due to the projective transformation, a one-to-one mapping between p_i and P_i is theoretically impossible. Therefore, a practically feasible approach is to predict the next position P_{t+1}, given the history of p_t, p_{t-1} ... , p_{t-m+1} , where m is the window size. This approach is segmentation-free, tractable, and can be

[1] http://www.vision.ee.ethz.ch/~calvin/calvin_upperbody_detector/

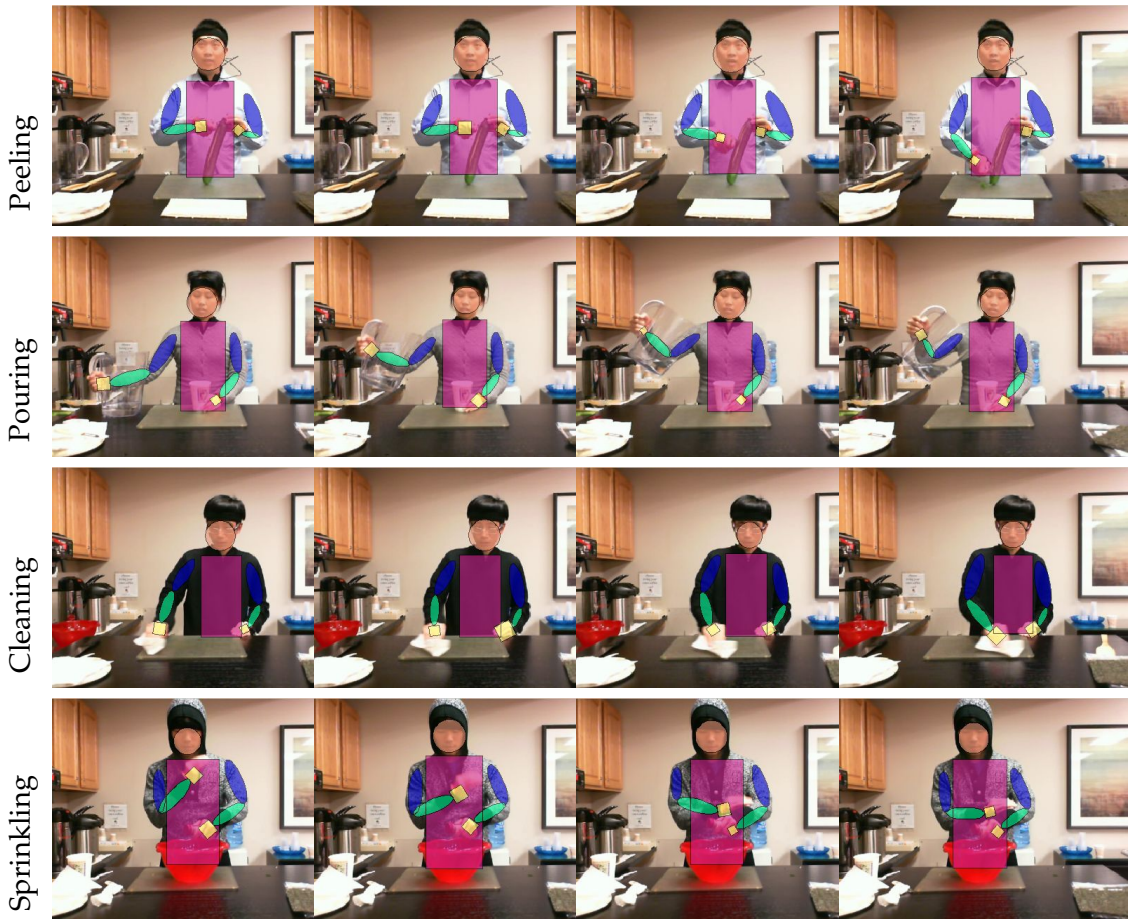


Figure 3.7: The results of the continuous pose localization. Four actions performed by four subjects were shown. Please refer to Fig. 3.1 and 3.4 for the coding of the colors.

implemented by many existing learning algorithms.

By concatenating the coordinates of the histories of p_i's and P_i's respectively, we create two matrices X and Y, where X is $n \times 4m$ and Y is $n \times 6$. Therefore, the problem can be formulated as follows:

$$Y = f(X) \tag{3.10}$$

where the unknown function $f(\cdot)$ is the mapping between the data points in the training set. This mapping is a multi-input-multi-output system, and can be approximated by linear equations in practice. In the chapter, we used Partial Least Squares (PLS) regression [Abd07] to estimate the function $f(\cdot)$.

3.3.3 Partial Least Squares in a Nutshell

Partial Least Squares regression is one of the statistical methods that handle Multivariate Multiple Linear Regression. Instead of finding linear coefficients between the responses and independent observations directly, it first projects the input variables and the observations to a reduced space using Principal Component Analysis (PCA), and then uses linear regression models to fit the parameters in the reduced space. Therefore, it attempts to use the major components of the observations to explain the major components of the responses. This method is particularly suited here when the matrix of observations is multicollinear. Fig. 3.8 illustrates this idea.

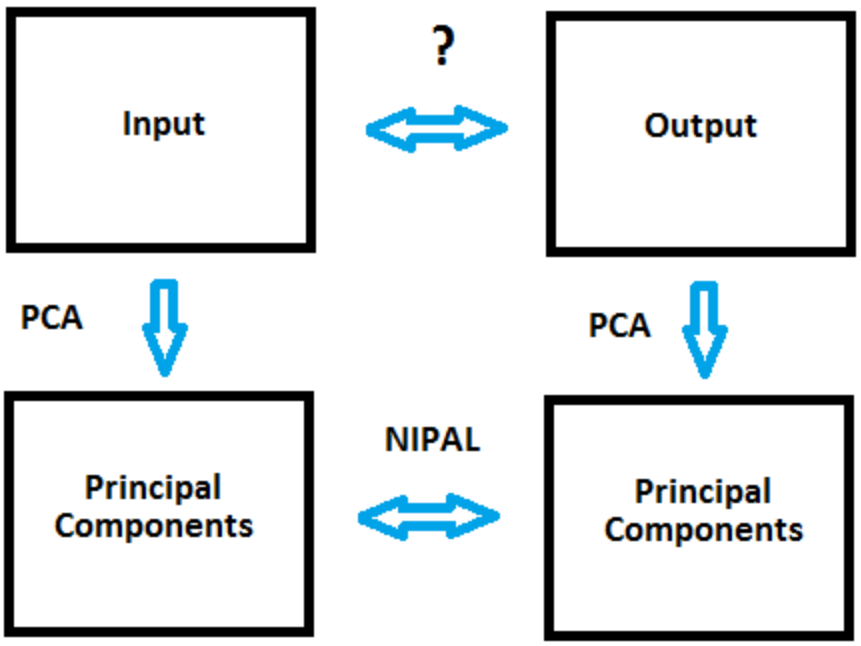


Figure 3.8: Illustrating the Partial Least Squares algorithm. Instead of modeling the linear relationship between the responses (output) and the observations (input) directly, it attempts to model the linear relationship between their principal components.

We briefly describe the procedure of PLS here. Given the number of the principal components p, PLS decomposes the zero-mean $n \times 4m$ matrix X and zero-mean $n \times 6$ matrix Y in Eq. 3.10 into:

$$X = TP^T + E \tag{3.11}$$

$$Y = Uq^T + e \tag{3.12}$$

where T and U are $n \times p$ matrices, P is a $2m \times p$ matrix, and the q is a $6 \times p$ matrix. The $n \times 2m$ matrix E and the $n \times 6$ matrix e are the residuals.

The PLS method then uses the nonlinear iterative partial least squares (NIPALS) algorithm to construct a set of weight vectors $W = \{w_1, w_2, ..., w_p\}$ such that

$$[cov(t_i, u_i)]^2 = \max_{|w|=1} [cov(Xw_i, Y)]^2 \tag{3.13}$$

where t_i is the i^{th} column of matrix T, u_i is the i^{th} column of matrix U, and $cov(\cdot)$ is the covariance. Please refer to [Abd07] and [SKHD09a] for details.

3.4 Experiments

The following four experiments demonstrate the usefulness of our approach.

First, we present a toy example that gives the intuition of the approach. Second, we show that the approach can effectively reconstruct the 3D poses using synthetic data, and it is robust to viewpoint changes. Third, we use real data from the UMD-Sushi and the YACD dataset to train and test the mapping module. We showed that PLS is superior to multiple linear regression on each response because it considers the covariance in the input and the output together. Finally, we demonstrate that the 3D reconstruction leads to better action recognition rates.

Learning the mapping amounts to computing the weights in Eq. 3.13 from the training data. We set the window size to 5 in all the experiments, and set the number of the principal components in the PLS method to 3.

3.4.1 Toy Example

We first present a toy example to demonstrate the intuition behind the proposed approach. The goal of this example is to recover the 3D coordinates from the 2D trajectories on a projection plane. In this example, four non-parametric nonlinear 3D curves were generated.

Then they were projected to a 2D plane using orthographic projection.

To make this example interesting, we used "CVPR", the four capital letters of this conference to generate the (x, y) coordinates of C_i $(i = 1, ..., 4)$ (shown as blue curves in the first row in Fig. 3.9), each of which is 1 pixel wide. The z coordinates of C_i were generated using the following parametric formula (blue curves in the second row in Fig. 3.9):

$$z = 50 \times sin(log\sqrt{x} + log\sqrt{y}) \tag{3.14}$$

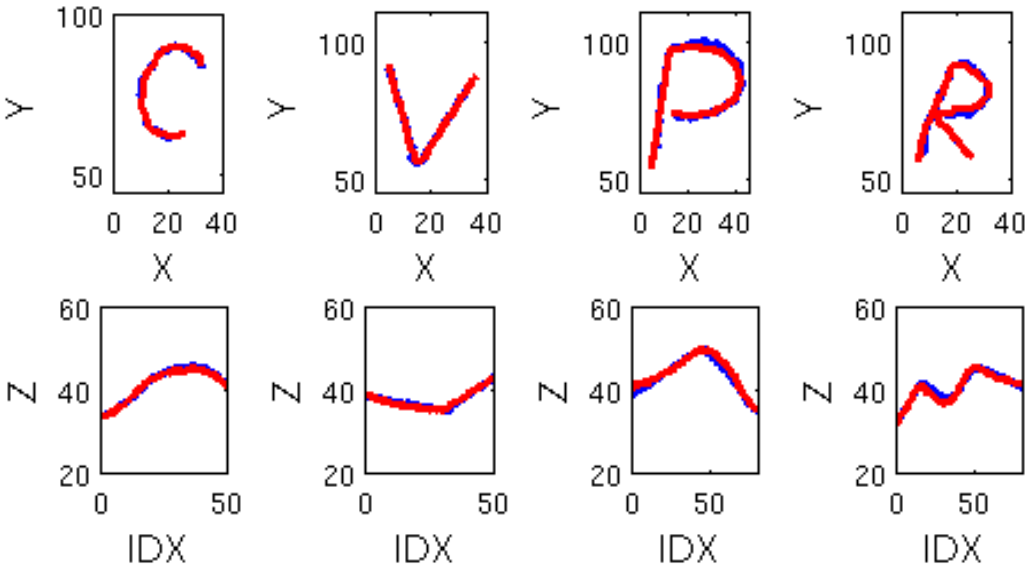


Figure 3.9: A toy example. We generated the (x, y) coordinates based on the letters "CVPR" (top, blue curves), and used Eq. 3.14 to generate the z coordinates (bottom, blue curves). The reconstructions in red overlap significantly with the blue curves.

We projected C_i $(i = 1, ..., 4)$ to a 2D plane using orthographic projection. Here, we chose the $viewpoint = (45°, 45°, 45°)$ as Azimuth, Elevation, and Roll of the WCS. The 3D trajectories were rotated and projected to the xy plane in the new space as the 2D projections.

To learn the mapping, we randomly selected 33% of the data points for training. Then, all the projection points were used to estimate the 3D positions for testing.

The red curves in Fig. 3.9 show the reconstruction in the original xy plane and the corresponding z coordinate, respectively. Referring to the figure, one can see that the blue curves and the red curves overlap significantly demonstrating that the reconstructions ap-

proximate the original 3D curves very well. The average reconstruction error is 0.84 units per data point. This example shows that the proposed approach is able to reconstruct different types of curves because the $C_i(x, y, z)$ ($i = 1, ..., 4$) are nonlinear.

3.4.2 Synthetic Example

The goal of the synthetic example is to show that the learned mapping is effective. Each MoCap sequence contains a few instances of the same action, and we projected the 3D positions of the body joints in the MoCap sequences to arbitrary 2D planes, generating synthetic 2D curves.

In this experiment, all the actions of subjects S_1 and S_2 were used for training and all the actions of other subjects for testing. To simplify the discussion, orthographic projection was used in this example, and the "Left Wrist" (LW) and the "Right Wrist" (RW) were used for demonstration.

Fig. 3.10 shows four examples of the original 3D trajectory and its reconstruction for the RW in the actions "Peeling" and "Pouring", when the $viewpoint = (45°, 45°, 45°)$. The start and the end poses of the right arm in these actions are also shown on top.

Comparing the same action performed by different subjects (blue curves in each column in Fig. 3.10), one can see the intra-class variance in 3D. Our mapping algorithm robustly handles the variance by modeling the process, and our reconstruction in the WCS (red curves) approximates the original trajectories (blue curves) reasonably well.

To statistically analyze the robustness of the module to change of camera viewpoint, we iterated all possible Azimuth and Elevation angles, while keeping the Roll zero. Then, for each viewpoint the average error was obtained by averaging the mean accumulated errors of each joint over all time frames.

Fig. 3.11a illustrates the average error in a color diagram. It shows that the expected error of each joint in each frame ranges from 2cm to 2.6cm for different viewpoints. Particularly, the error map has some local minima, which may indicate that the actions in our YACD dataset are best approximated and understood in 3D from these viewpoints. For instance, the viewpoint near $45°$ in elevation and $0°$ in azimuth has the best fitting results, which corresponds to the normal watching angle in commercial cooking shows.

Furthermore, we would like to demonstrate that our approach is robust when the test viewpoint is *different* from the training viewpoint. Given a viewpoint for training, we

changed the test viewpoint to a slightly different angle, and use the trained model to predict the 3D data.

Fig. 3.11b shows the error for change in viewpoint between -10° to 10°. The error is the average from a number of viewpoints. The result suggests that the reconstruction is accurate when there is small change of viewpoint, and degrades reasonably when the change becomes large.

This synthetic experiment demonstrates that the PLS can recover the 3D position from arbitrary projections accurately, and its performance is robust to reasonable changes in viewpoint between the training and testing samples.

3.4.3 3D Reconstruction on Real Data: A Pilot Study

We carried out a pilot study in this section on the UMD-Sushi dataset. In this pilot study, the body joints can be easily tracked. Therefore, the 3D reconstruction can be analyzed more intuitively.

We first detect the trajectories of four body joints, namely "Left Wrist", "Left Elbow", "Right Wrist", and "Right Elbow", in videos, and use the results to demonstrate the usefulness of the mapping from 2D to 3D.

Fig. 3.12 shows the trajectories (yellow curves) of the right wrist for six different actions in our UMD-Sushi dataset. Three instances performed by the same subject were visualized.

One can see in Fig. 3.12 that the 2D positions of the joints in the different actions may be similar, but their trajectories are different. This suggests that using motion history is sufficient to distinguish the actions and estimate the 3D positions.

In this example, all actions of four body joints from two subjects were used for training and the remaining subject's data for testing.

First, we low-pass filtered the trajectories in 2D to reduce noise, and selected the "active" trajectories with the range of trajectories larger than a threshold. Then, we applied the proposed algorithm to the active trajectories in 2D and in 3D for each body joint.

Fig. 3.13 illustrates the result. The blue curves denote the original 3D curves in the WCS and the red curves denote the reconstructed results using the 2D trajectories on unknown (but fixed) camera coordinate system (Bottom, Fig. 3.13). Two instances in the visual domain and in the motoric domain are shown (Top, Fig. 3.13). The blue skeleton denotes the original MoCap data, and the red skeleton denotes the reconstructed 3D positions.

Subject	S_1	S_2	S_3	S_4
PLS (cm/f)	8.28	9.48	11.62	11.69
MR (cm/f)	8.61	10.68	12.51	12.96

Table 3.1: Performance comparison between the Partial Least Square (PLS) and the Multivariate Regression (MR). The performance was measured by the averaged standard deviation for each subject.

3.4.4 3D Reconstruction on the YACD Dataset

Real data from the camera and the MoCap suit were used in this experiment. We first localized the upper body parts using the method described in Sec. 3.3.1, and then mapped the 2D trajectories from to 3D.

In the following examples, we performed manually a temporal segmentation on the YACD, and normalized the data in each action segment to 50 frames.

3.4.4.1 Computing Body Parts in Videos

Our efficient pose localization can detect poses effectively in videos. Fig. 3.14 shows 16 examples of our pose estimation on the YACD dataset. The first frame of each action instance and the corresponding trajectory of the RW are shown. Referring to figure, one can see that the viewpoint of subject S_4 is slight different from those of other subjects, because we intentionally changed the angle during recording, to evaluate the robustness of the approach.

Fig. 3.14 demonstrates that the 2D visual processing is robust across subjects. The smooth trajectories of body joints further enable the processing using time series techniques.

3.4.4.2 Learning the Mapping from Videos to MoCap

All actions of subjects S_1 and S_2 were used for training and all four subjects' data were used for testing.

First, we low-pass filtered the trajectories in 2D to reduce noise. Then, we applied the proposed algorithm to the smoothed trajectories in 2D and in 3D for each body joint.

Table 3.1 shows the performance measured by the standard deviation of error between the reconstruction and the ground truth (MoCap data). Compared to Fig. 3.11, the experiments on real data have higher errors than the synthetic data on average. The ap-

proximations on novel subject data (S_3 and S_4 in Table 3.1) also have reasonably larger error (10%-15%) compared to the training set. In addition, S_4 has a marginally larger error compared to S3 because of the slight viewpoint change.

We further compared the PLS method to Multivariate Regression (MR). As can be seen from the table the average error of PLS is smaller. This is because PLS considers a linear relationship between the covariances instead of the original data. Therefore, it models the process better.

Because of the loss of the third dimension due to the projective transformation, and errors in the 2D pose localization, the reconstruction cannot be perfect. We show how the mapping facilitates action recognition next.

3.4.5 3D Action Recognition

We demonstrate that the 3D mapping facilitates action recognition in the following experiments. Two different schemes of partitioning the data were used for training and testing, and the performance was compared against a 2D action recognition algorithm based on trajectories.

In the first scheme, we partition the YACD dataset by subjects. All ten actions performed by a randomly chosen subject were used for training, and the rest for testing. In the second scheme, four randomly chosen actions performed by all subjects were used for training and the remaining six actions from all subjects were used for testing.

3.4.5.1 Partition the YACD by Subjects

In this scheme, one subject, S_i $(i = 1..4_i)$, was randomly chosen for training the mapping, and all the actions of the remaining three subjects were used in a test set TS_i.

As a baseline, the algorithm in [MPK] was used. A 5-fold cross validation on each TS_i was performed. Fig. 3.15a shows the averaged confusion matrix.

We then chose two classifiers, Naive Bayes (NB) and BayesNet (BN), and performed a 5-fold cross validation on each TS_i using the original 2D trajectories and the reconstructed 3D trajectories, respectively. The averaged confusion matrices are shown in Fig. 3.16.

The recognition rate of NB on the 2D trajectories is smaller than the baseline because we only considered the 2D joints as opposed to a large number of tracked keypoints. However, our 3D representation outperformed the base line descriptor 7.3% because we directly

modeled underlying motion process. One can also see that the reconstructed 3D data are preferable because they facilitated both classifiers and increased the average accuracy from 60% to 77% (NB), and 74% to 82% (BN), respectively.

This experiment demonstrates that the mapping recovers the 3D locations of seen actions performed by novel subjects, and further improves the action recognition accuracy.

3.4.5.2 Partition the YACD by Actions

The evaluation procedure was similar to the one in Sec. 3.4.5.1 except that we partitioned the actions instead of the subjects. Four actions performed by all subjects were randomly chosen for training the mapping module, and the other six actions were used for testing.

To evaluate the performance, we ran the above procedure 20 times, and averaged the accuracy for each action. Fig. 3.15b plots the results of the two classifiers on 2D and 3D inputs, respectively. Clearly, the info in the recovered 3^{rd} dimension improved the performance of both classifiers. For instance, the increase was significant for the actions "Cutting" and "Flipping" where the 2D information may not be sufficient. On average, we obtained 4.8% (NB) and 11% (NB) improvement using the 3D reconstruction, respectively.

This experiment suggested that we can use the 3D information of the seen actions to predict novel actions.

3.5 Chapter Conclusion

We have presented a visual motoric mapping for action representation, which we obtained using a statistical learning method called Partial Least Squares. The method involves detecting the body joints, and then mapping the 2D joint trajectories in visual space to 3D motion in motoric space. Two dataset consisting of eighteen cooking actions performed by four subjects in a kitchen environment were collected. Experiments on synthetic and real data showed that our method robustly represents human actions.

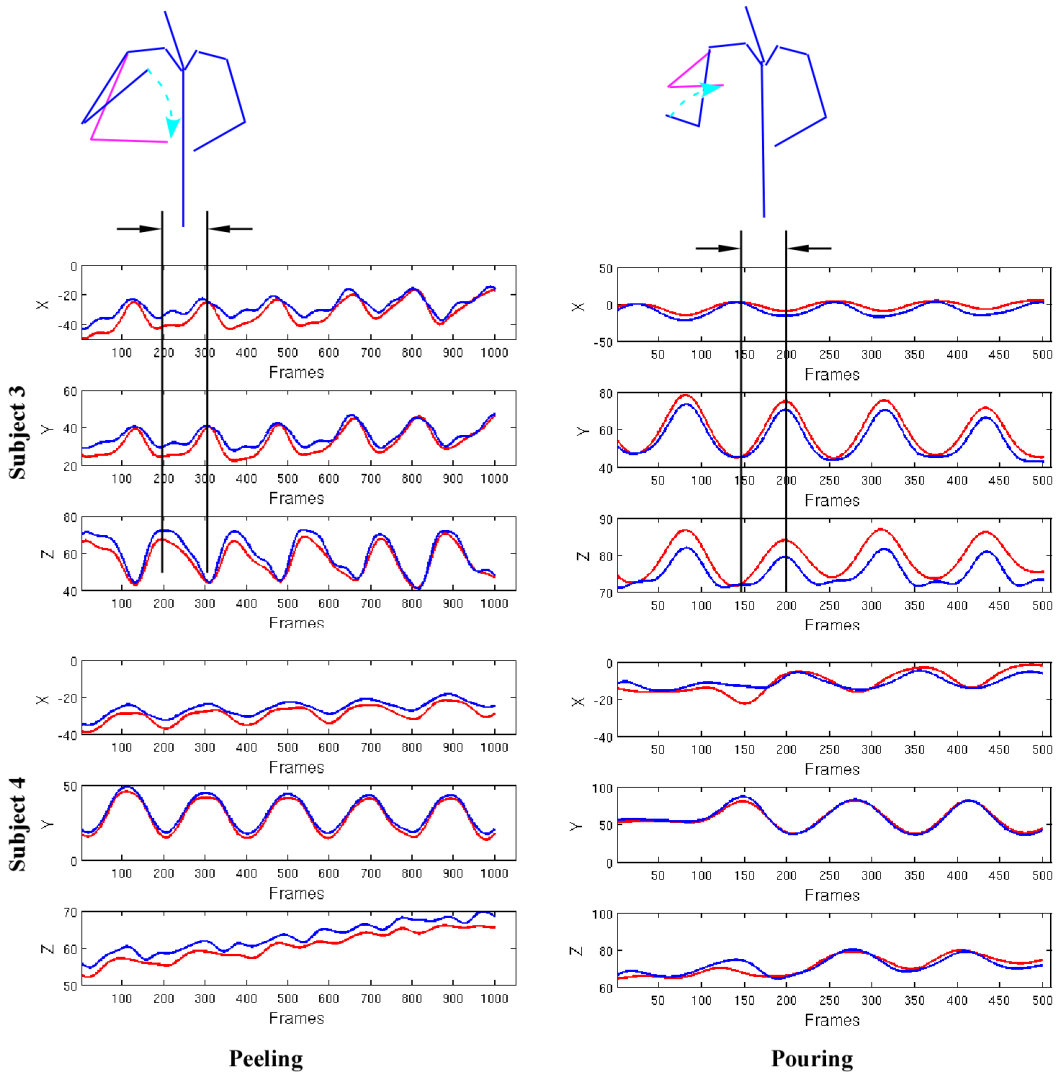
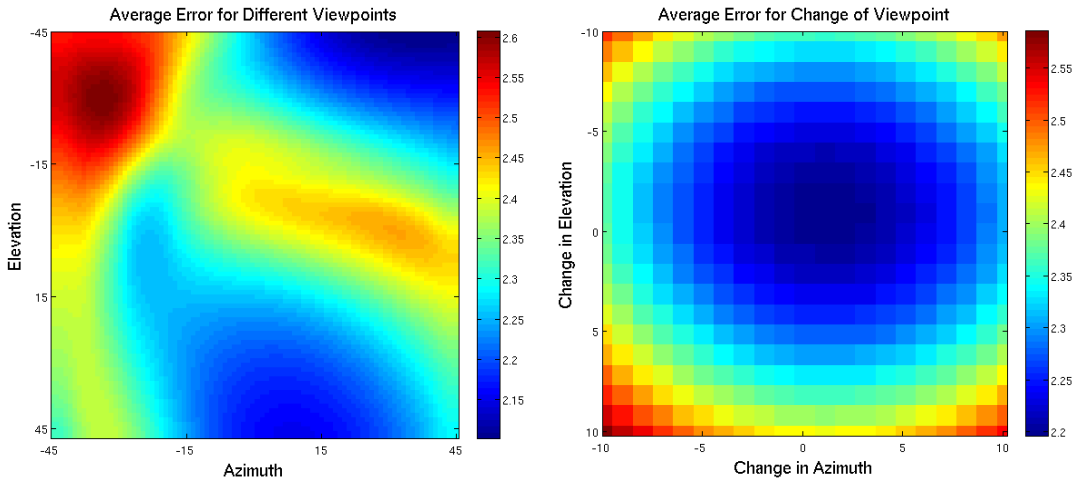


Figure 3.10: Synthetic 3D pose reconstruction of "Right Wrist" (RW) in "Peeling" and "Pouring" on the YACD dataset. Blue curves: the original 3D curves in WCS. Red curves: the reconstructed results with the 3D rotation angle $(45°, 45°, 45°)$. The start pose (blue) and the end pose (pink) of the right arm in an instance are also shown on top. The dashed arrow denotes the motion.



(a) (b)

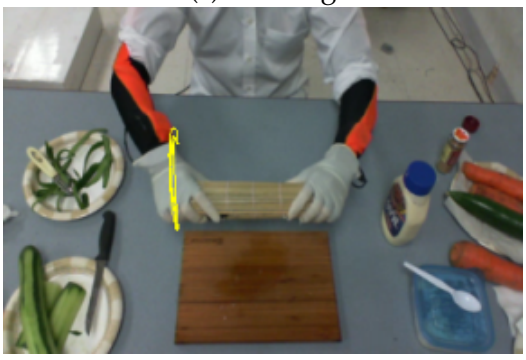
Figure 3.11: Robustness test on synthetic data. The 3D trajectories of two joints {RW, LW} were projected to 2D planes using orthographic projection. (a) The reconstruction accuracy for viewpoints (Azimuth and Elevation). (b) For each pair of Azimuth and Elevation, we changed the testing angle between -10° to 10°. The error is the average from the number of viewpoints.

(a) Peeling

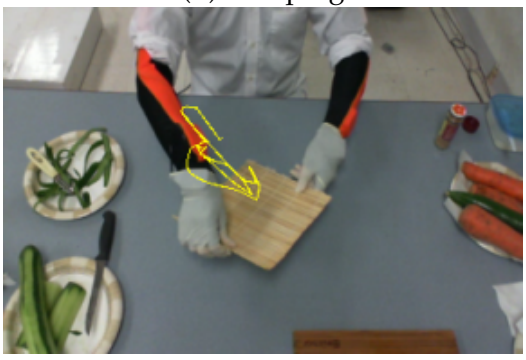
(b) Pickup

(c) Pressing

(d) Scooping



(e) Transferring



(f) Turning

Figure 3.12: Computing 2D joint trajectories from videos. We showed the trajectories of three instances per action for six different actions.

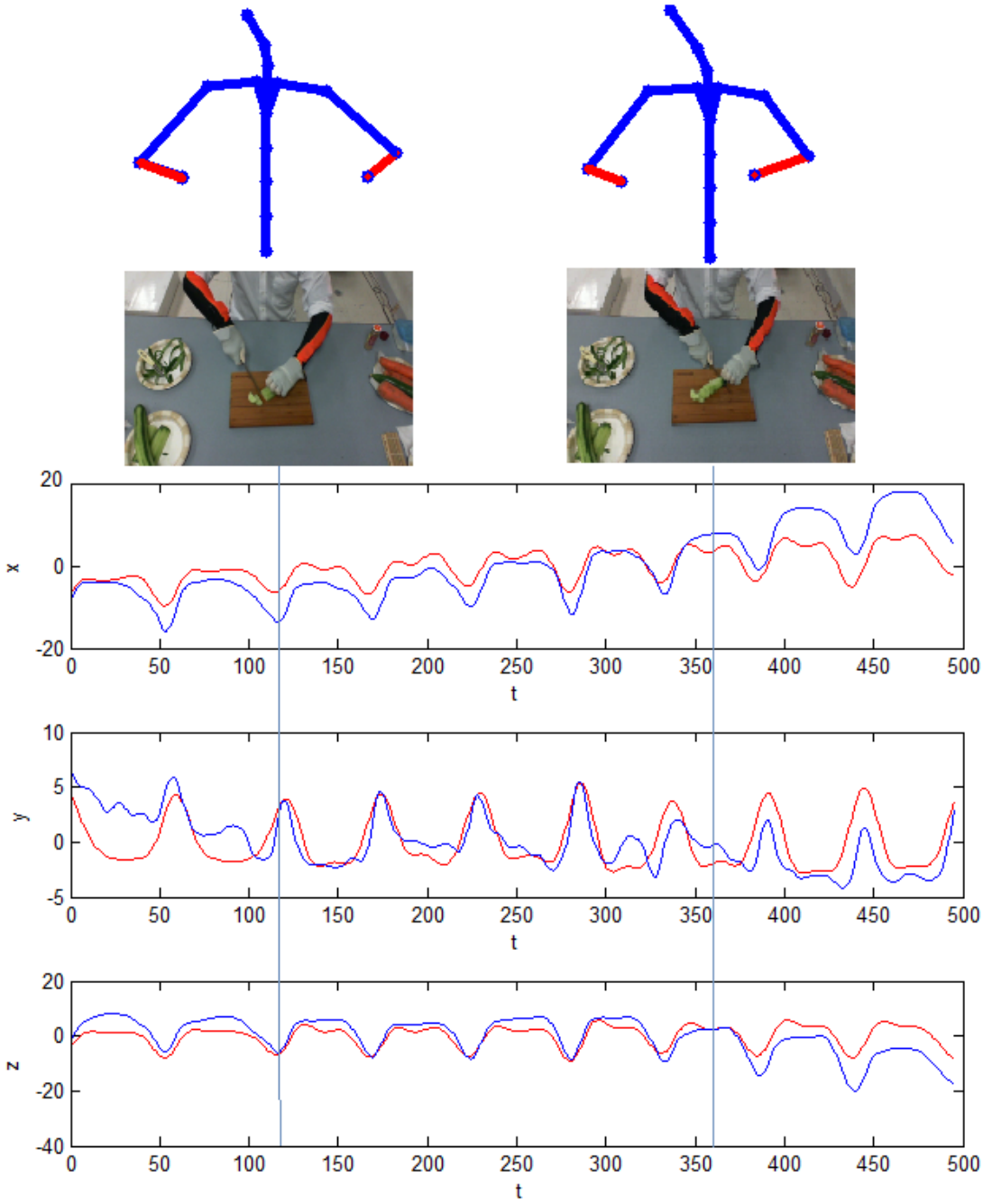


Figure 3.13: Illustrating the proposed approach using real data. Bottom: the blue curves denote the original 3D curves in the WCS and the red curves denote the reconstructed results using the 2D trajectories in unknown (but fixed) camera coordinate system. Top: two instances in the visual domain and the motoric domain are shown. Blue skeleton: the original MoCap data; Red skeleton: the reconstructed 3D positions.

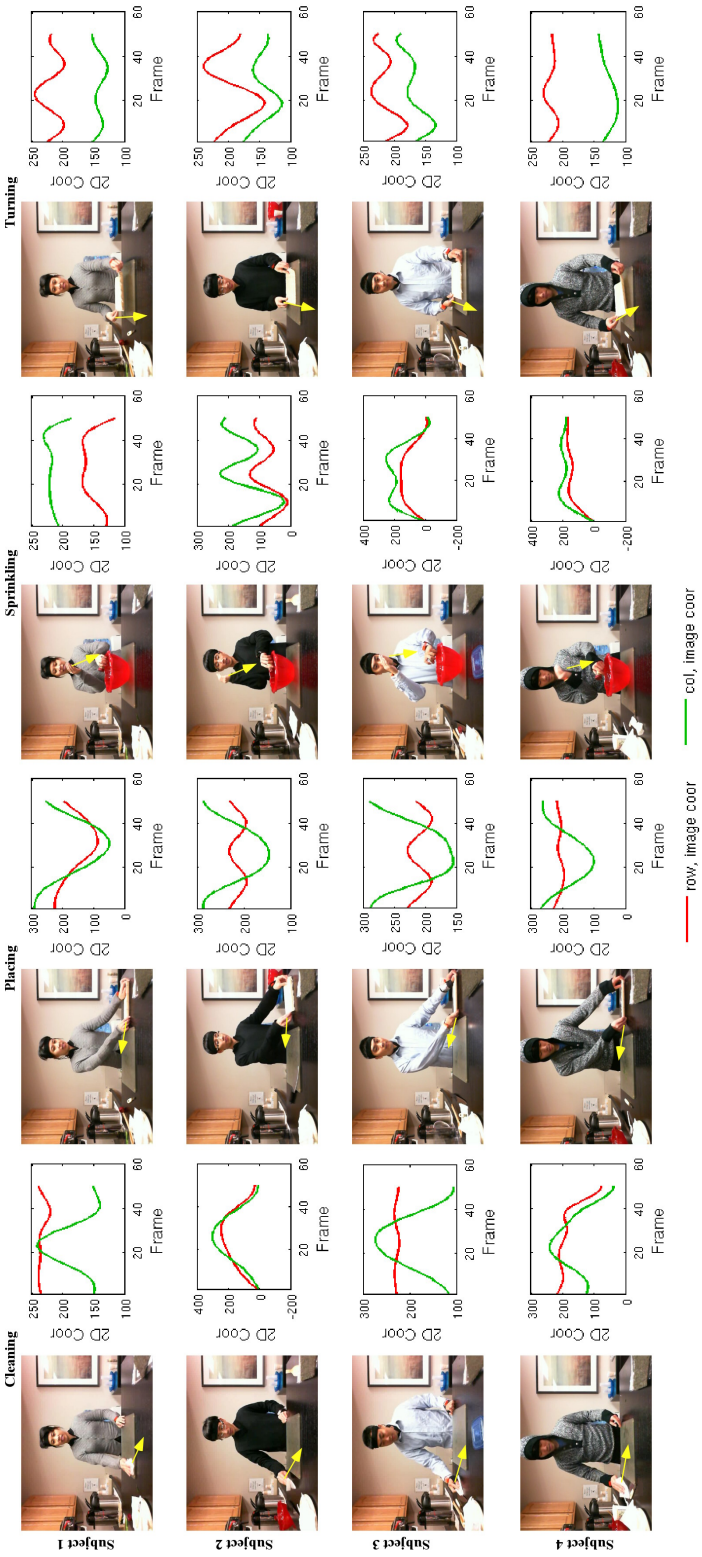
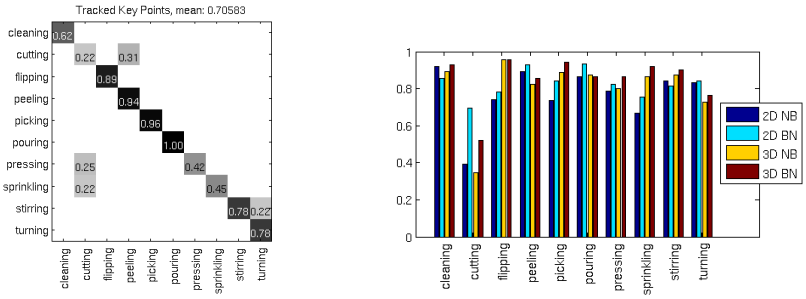


Figure 3.14: Results for pose localization. For each instance the first frame and the 2D joint trajectory of the "Right Wrist" (RW) in the videos are shown. The yellow arrows denote the motion direction. The viewpoint of subject S_4 is slightly different from others.



(a) (b)

Figure 3.15: (a) Confusion matrix of the baseline [MPK] (Naive Bayes) on our dataset. (b) Accuracy of Naive Bayes (NB) and BayesNet (BN) applied to the 2D trajectories and the 3D reconstruction.

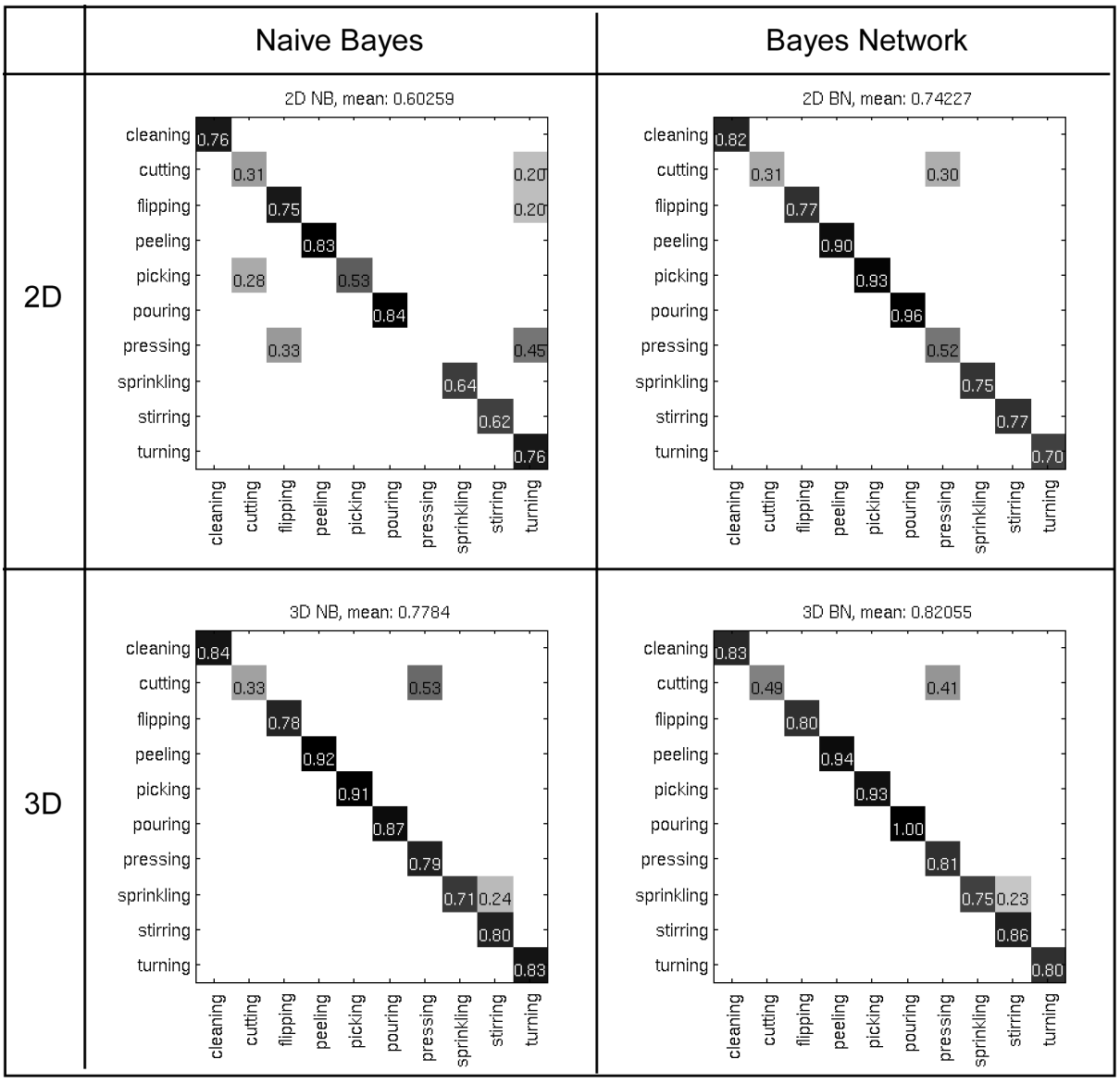


Figure 3.16: Confusion matrices of Naive Bayes (NB) and BayesNet (BN) applied to the 2D trajectories and the 3D reconstruction.

Chapter 4

Learning Shift-Invariant Sparse Representation of Actions

4.1 Chapter Summary

A central problem in the analysis of motion capture (MoCap) data is how to decompose motion sequences into primitives. Ideally, a description in terms of primitives should facilitate the recognition, synthesis, and characterization of actions. We propose an unsupervised learning algorithm for automatically decomposing joint movements in human motion capture (MoCap) sequences into shift-invariant basis functions. Our formulation models the time series data of joint movements in actions as a sparse linear combination of short basis functions (snippets), which are executed (or "activated") at different positions in time. Given a set of MoCap sequences of different actions, our algorithm finds the decomposition of MoCap sequences in terms of basis functions and their activations in time. Using the tools of L_1 minimization, the procedure alternately solves two large convex minimizations: Given the basis functions, a variant of Orthogonal Matching Pursuit solves for the activations, and given the activations, the Split Bregman Algorithm solves for the basis functions. Experiments demonstrate the power of the decomposition in a number of applications, including action recognition, retrieval, MoCap data compression, and as a tool for classification in the diagnosis of Parkinson (a motion disorder disease).

This chapter is based on the paper appeared in the IEEE Conference on Computer Vision and Pattern Recognition 2010. Please refer to [7] in Appendix A for details.

4.2 Introduction

Interpreting human behavior is a newly emerging area that has attracted increasing attention in computer vision. One of the intellectual challenges in modeling human motion is to come up with formalisms for describing and recognizing human actions in motion capture (MoCap) sequences. Fundamentally, the primitives should assist the recognition, synthesis, and characterization of human actions. From this perspective, the formalism of the primitives is essential to action representation.

Human actions by their nature are sparse both in action space domain and time domain. They are sparse in action space, because different actions share similar movements on some joints, and also different joints share similar movements. They are sparse in the time domain, because we do not want much overlap of the individual movements on a single joint. These observations make the concept of shift-invariant sparse representation as the primitives of human actions very attractive, where shift invariant means that the output does not depend explicitly on time, *e.g.*, the same action can have multiple realizations at different times.

Let us get into more detail. We are given many MoCap sequences. The data from a motion capture suit are time series of three rotation angles each at a number of joints on the human body. Each of these sequences consists of a number of instances of different actions (where an instance of an action could be a step of a "running" sequence, or a single "kick", or "jump"). Our goal is to obtain from these action sequences a set of basis functions that could be used for approximating the entire set of the actions.

Our basis functions are chosen to be smooth functions and about the length of an instance of an action (Fig. 4.1). This enables us to achieve a useful underlying representation of different actions. The joint movements in an instance of an action are approximated by a sparse linear combination of basis functions ("Action Unit" in Fig. 4.1). To achieve a meaningful behavioral interpretation the weights are defined to be positive. Multiple instances of the same joint movement are realized by executing (or "activating") the linear combination of basis functions at different instances of time, but with different strength ("Activation" in Fig. 4.1). That is, all basis functions involved in the representation of a single joint are activated simultaneously. But different joints are activated separately.

Our action representation then is the weights of the basis functions along with their

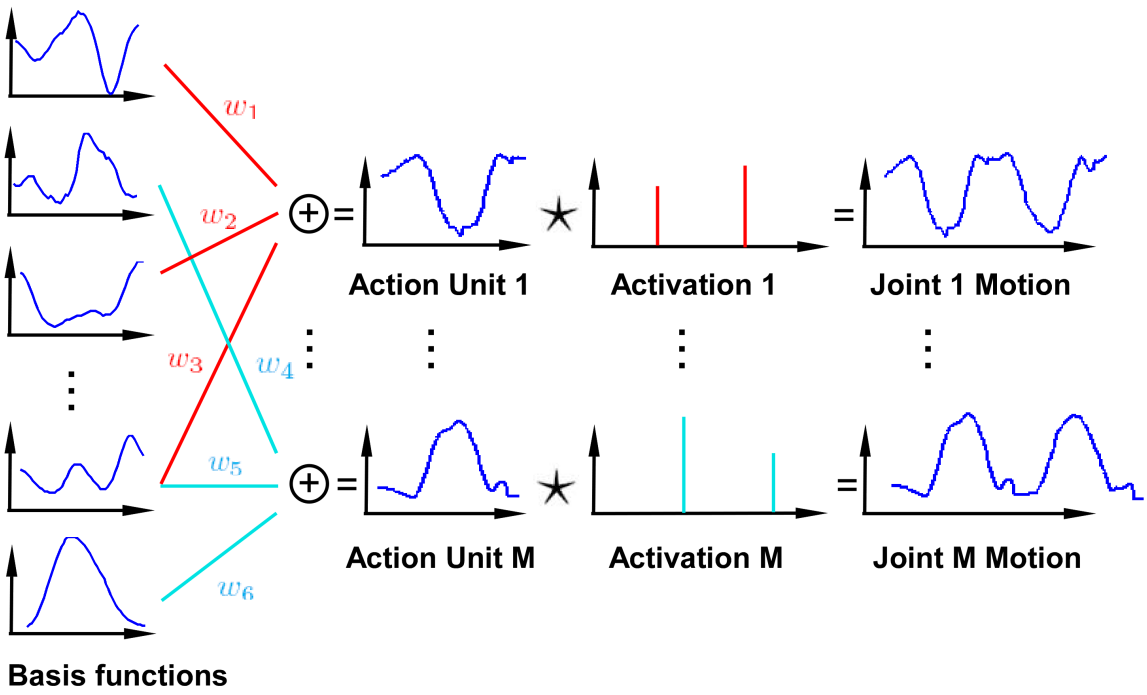


Figure 4.1: Modeling human motion in MoCap sequences using shift-invariant sparse representation. The short basis functions are sparsely selected and linearly combined to create action units for individual joints. The units may be shifted to different locations where multiple instances of the movement are realized. The time shift is modeled by the convolution (denoted by $\star$).

activations. Once these shift-invariant basis functions are learned, we can approximate any novel action sequence using a weighted combination of a number of these basis functions.

In our learning procedure, we solve for both the basis functions of the actions and the times when these functions are "activated". Solving them together would amount to a complicated non-convex optimization with a large number of variables. However, the optimization problem is convex in either the basis functions or the activations. Our method thus solves alternately for the two set of parameters. Recently developed L_1 minimization techniques allow us to solve these two problems effectively. Given a set of basis functions, a variant of Orthogonal Matching Pursuit [TG07] is used to obtain the activations by solving a non-negative L_1 minimization problem with a large number of variables. Given the activations, the Split Bregman Algorithm [GO09] is used to solve an L_1 regularized liner least square problem.

The characteristics of our decomposition approach are:

1. Our unsupervised algorithm learns a high-level sparse representation (the primi-

tives) of action which allows recognizing actions in MoCap sequences effectively.

2. The shift-invariant modeling naturally handles the MoCap sequence composed of multiple instances of different actions.

3. The sparse activations explicitly express the coordination among different joints.

The rest of the chapter is organized as follows. Sec. 4.3 presents the algorithm for learning the basis functions. Sec. 4.4 summarizes an algorithm used for normalizing the length of MoCap sequences. Sec. 4.5 demonstrates the usefulness of our action representation on four applications, and Sec. 4.6 concludes the chapter.

4.3 Shift-Invariant Sparse Modeling of Actions

A MoCap sequence consists of the time series of three rotation angles each at a number of joints on the human body. In our approach, we approximate each time series by sparse linear combinations of shift-invariant sparse features.

For simplicity, we start our journey from the following example. A body joint rotation s ($1d$ time series) consists of the movements of multiple instances of the same action. Given N short basis functions b_i ($i = 1, 2, ...N$) which have the length about an instance of an action, we would like to approximate s as follows:

$$s \approx \sum_i a \star w_i b_i, \tag{4.1}$$

where a is the sparse activation for s, and $\star$ is the convolution operator (Fig. 4.1). The variables a and w_i are non-negative. Eq. 4.1 is equivalent to

$$\begin{aligned} s &\approx \sum_i w_i a \star b_i \\ &= \sum a_i \star b_i, \end{aligned} \tag{4.2}$$

where $a_i = w_i a$. This means we can model the shift of each individual basis function separately, with the additional constraint that all the activations a_i ($i = 1, 2, ...N$) must have non-zero values at the same time when used for approximating s.

In the following formulation, we first discuss a solution for Eq. 4.2 in Sec. 4.3.1 and 4.3.2. The additional constraint is enforced when we solve the activations in Sec 4.3.3.1.

4.3.1 Problem Formulation

Given a set of M $1d$ signals s_j, $j = 1, 2, ...M$, each of which is of length l and represents a time series of a joint movement, we want to approximate all s_j as the convolution between the activations and the basis functions, *i.e.*,

$$s_j = \sum_i a_i^j \star b_i + n_j, \tag{4.3}$$

where a_i^j ($j = 1, 2, ...M$, $i = 1, 2, ...N$) are the sparse non-negative activations for the i^{th} basis function in the j^{th} signal, and n_j is the noise.

We enforce that the activations are sparse, and the basis functions are sparse in the Fourier domain. Therefore, the modeling poses the following L_1 regularized optimization problem:

$$\min_{(a_i^j, b_j)} \sum_j |s_j - \sum_i a_i^j \star b_j|_2 + \mu_1 \sum_{i,j} |a_i^j|_1 + \mu_2 \sum_i |F\hat{b_i}|_1, \tag{4.4}$$

where $|\cdot|_p$ is the L_p norm of the vector, F is the Fourier transform matrix, and $\hat{b_i}$ are the zero-padded b_i which are of length l.

Solving the activations and the basis functions together would amount to a non-convex optimization with a large number of variables. In Sec. 4.3.2, we re-formulate the problem in the frequency domain.

4.3.2 Formulating the Problem in Frequency Domain

We show that the optimization problem is convex in either the basis functions or the activations. Therefore, a coordinate descent algorithm is used to alternately solve two large convex L_1 regularized problems.

The convolution in time domain is equivalent to the dot product in frequency domain. Therefore, Eq. 4.3 is equivalent to:

$$S_j \approx \sum_i A_i^j \cdot \hat{B}_i \tag{4.5}$$

where $\cdot$ is the pairwise multiplication operation, $A_i^j = Fa_i^j$, and $\hat{B}_i = F\hat{b_i}$, respectively.

Denoting $\overline{X}$ as the square matrix whose diagonal is X, we have:

$$A_i^j \cdot \hat{B}_i = \overline{A_i^j}\hat{B}_i = \overline{\hat{B}_i}A_i^j \tag{4.6}$$

Therefore, Eq. 4.5 is equivalent to

$$\begin{aligned} S_j &\approx \begin{bmatrix} \overline{\hat{B}_1} & \dots & \overline{\hat{B}_N} \end{bmatrix} \begin{bmatrix} A_1^j \\ \vdots \\ A_N^j \end{bmatrix} \\ &= \begin{bmatrix} \overline{\hat{B}_1} & \dots & \overline{\hat{B}_N} \end{bmatrix} \begin{bmatrix} F & & \\ & \ddots & \\ & & F \end{bmatrix} \begin{bmatrix} a_1^j \\ \vdots \\ a_N^j \end{bmatrix} \\ &= \mathcal{B}a_j, \end{aligned} \tag{4.7}$$

where $a^j = [a_1^j; ...; a_N^j]^T$, and

$$\mathcal{B} = \begin{bmatrix} \overline{\hat{B}_1} & \dots & \overline{\hat{B}_N} \end{bmatrix} \begin{bmatrix} F & & \\ & \ddots & \\ & & F \end{bmatrix}.$$

Similarly, Eq. 4.5 can be rewritten as:

$$\begin{aligned} S_j &\approx \begin{bmatrix} \overline{A_1^j} & \dots & \overline{A_N^j} \end{bmatrix} \begin{bmatrix} \hat{B}_1 \\ \vdots \\ \hat{B_N} \end{bmatrix} \\ &= \begin{bmatrix} \overline{A_1^j} & \dots & \overline{A_N^j} \end{bmatrix} \begin{bmatrix} F & & \\ & \ddots & \\ & & F \end{bmatrix} \begin{bmatrix} \hat{b_1} \\ \vdots \\ \hat{b_N} \end{bmatrix} \\ &= \begin{bmatrix} \overline{A_1^j} & \dots & \overline{A_N^j} \end{bmatrix} \begin{bmatrix} F_l & & \\ & \ddots & \\ & & F_l \end{bmatrix} \begin{bmatrix} b_1 \\ \vdots \\ b_N \end{bmatrix} \\ &= \mathcal{A}^j \mathcal{F} b, \end{aligned} \tag{4.8}$$

where $\mathcal{A}^j = [\overline{A_1^j}, ..., \overline{A_N^j}]$, $b = [b_1; ...; b_N]^T$, the matrix F_l as the first l columns of the F, and

$$\mathcal{F} = \begin{bmatrix} F_l & & \\ & \ddots & \\ & & F_l \end{bmatrix}.$$

Let S, $\mathcal{A}$, a and b denote the concatenations of all possible S_j, $\mathcal{A}^j$, a^j and b_i in the column form. Eq. 4.4 is convex in either a or b, so we solve it alternately as two convex optimization problems.

Given the basis functions b in time domain, from Eq. 4.4 we obtain:

$$\min_a |S - \mathcal{B}a|_2 + \mu_1 |a|_1 \tag{4.9}$$

Eq. 4.9 is the sum of M independent subproblems, all of which are convex. We can approximate them separately.

Given the activations a, from Eq. 4.4 we obtain:

$$\min_b |S - \mathcal{A}\mathcal{F}b|_2 + \mu_2 |\mathcal{F}b|_1 \tag{4.10}$$

Both Eq. 4.9 and 4.10 are convex. Therefore, the objective function Eq. 4.4 is always non-increasing using the updates. Eq. 4.9 is solved using the Orthogonal Matching Pursuit, and Eq. 4.10 is solved using the Split Bregman iterative algorithm. To avoid trivial results, we normalize the basis function in amplitude in each iteration.

4.3.3 Solving the Problem using L_1 Minimization

4.3.3.1 OMP for Solving the Activations

We use a variant of the Orthogonal Matching Pursuit (OMP) to solve Eq. 4.9. The variant amounts to implementing Orthogonal Matching Pursuit in a batch mode.

Orthogonal Matching Pursuit is a greedy algorithm. It progressively picks the new basis which minimizes the residual. The major advantages of this algorithm are its ease of implementation and its speed. This approach can easily be extended to related problems, such as finding non-negative bases [DT05b].

In our modeling, a single body joint movement is a sparse combination of the basis

functions, with the weights non-negative and the additional constraint that the activations must be coherent. This is solved as follows: Given the movement ($1d$ time series) of a joint, we progressively pick the new basis at the locations found in the previous steps, and minimize the residual of all the time series. We enforce the solution to be positive by checking which basis to choose and checking the weights found in the least-squares minimization.

During the optimization, a sparse subset of basis functions is automatically selected. In our implementation, we allow a maximum of 4 basis functions at a single activation at one joint, with the total number of basis functions being 15. This makes it easier to compare the weights of the same joint in action retrieval and classification.

4.3.3.2 Split Bregman Algorithm for Solving the Bases

As defined in the literature, the Split Bregman Iterative Algorithm is applied to the following problem

$$\min_{u} J(u) + H(u), \tag{4.11}$$

with $u \in R^n$, $J(u)$ is the L_1 norm of a function of u and is continuous but not differentiable function, and $H(u)$ is the L_2 norm of a function of u and is continuous differentiable. In our case, $J(u) = |\mathcal{F}b|_1$ and $H(u)$ is the L_2 norm of the approximation error.

By introducing $|d - \phi u|_2$ and $E(u, d) = |d|_1 + H(u)$, we rewrite Eq. 4.11 as

$$\min_{(u,d)} E(u, d) + \lambda/2|d - \phi u|_2. \tag{4.12}$$

The solution is given by iteratively updating the following three equations:

$$u^{k+1} \leftarrow arg\min_{u} H(u) + \lambda/2|d^k - \phi u - p^k|_2 \tag{4.13}$$

$$d^{k+1} \leftarrow arg\min_{d} |d|_1 + \lambda/2|d - \phi u^{k+1} - p^k|_2 \tag{4.14}$$

$$p^{k+1} \leftarrow p^k + \phi u^{k+1} - d^{k+1}$$

This "splits" Eq. 4.12 into the subproblems. Eq. 4.13 is a 2^{nd} order continuous differential function that can be solved efficiently. Eq. 4.14 is solved by shrinkage operation[1].

[1] $shinkage(x, y) = sgn(x)\max(|x| - y, 0)$.

By the alternately update in the Split Bregman Algorithm, we obtain the optimal sparse solution for Eq. 4.10.

4.4 Preprocessing: Normalization for Handing Actions with Various Speeds

It is important to handle action sequences of different speeds. For this we use our action segmentation algorithm[2]. This algorithm breaks an action sequence into action segments. We then compute the average length of the action segments and use it to normalize the sequence.

Our goal is to find the discontinuities in the 3^{rd} order derivative of the time series. Motivation for this approach comes from the work of d'Avella *et al.* [ddSB03], who found that the change of the muscle force indicates the time of action change, and the change of muscle force is proportional to the 3^{rd} order derivative of the time series.

Our algorithm partitions a MoCap sequences by minimizing the sum of the pairwise distances of the envelope extrema of the different joints. In this algorithm, we use the quaternion representation for rotation.

The quaternion series of a certain joint is a 4D vector

$$\mathbf{X}(t) = [x_1(t), x_2(t), x_3(t), x_4(t)]^T \tag{4.15}$$

The jerk of $\mathbf{X}(t)$ is computed as

$$J(t) = |\frac{d^3(\mathbf{X}(t))}{dt^3}|_2 \tag{4.16}$$

To minimize the error in computing the derivative, we smooth the data using a low pass filter.

To measure the jerk better, we compute the jerk envelope

$$\text{Env}(t) = |\text{Hilbert}(J(t))|_2 \tag{4.17}$$

for every joint, where Hilbert$(\cdot)$ is the Hilbert transform. This is a standard approach for computing the signal envelope [Bra99]. Then we process Hilbert$(\cdot)$ using as a low pass

[2]Please refer to Chap. 6 for detailed analysis of the temporal segmentation of human actions.

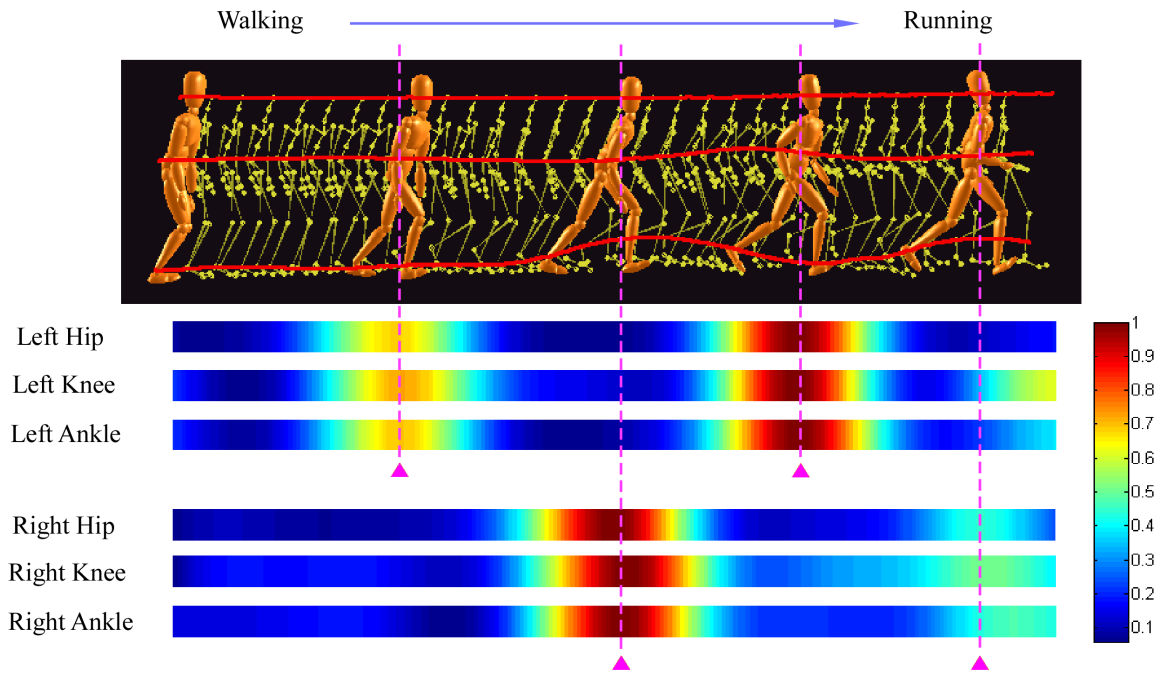


Figure 4.2: Estimating the average action speed by measuring the action discontinuities. A sequence "walking to running" from the University of Bonn dataset [MRC+07] is shown. The poses corresponding to the discontinuities are displayed as mannequin. The trajectories of the head, the left elbow, and the right ankle are drawn in red.

filter a Butterworth filter, and compute the envelope's extrema of the filtered Env(t) for every joint.

Fig. 4.2 gives an example of the speed-varying action "walking to running" from the University of Bonn dataset [MRC+07]. Here, we show the jerk envelopes of the joints "Hip", "Knee", and "Ankle" of both legs, respectively. The envelopes are color coded and normalized in magnitude. The breaking poses (mannequins) can be selected by finding the optimal "alignment" of the envelope extrema of the different joints (the purple dash lines in Fig. 4.2).

The alignment can be solved as an optimization problem using the envelope extrema of all joints. It is applied until the average pairwise distances is larger than a threshold ϵ.

4.5 Experiments

The following four experiments demonstrate the usefulness of our representation: First, we present the basis functions learned from our own dataset in Sec. 4.5.2. Second, we show that our basis functions are well suited for approximating novel actions. This allows

us to substantially compress novel MoCap data (Sec. 4.5.3). Third, the experiments in Sec. 4.5.4 demonstrate that using only the magnitude of the activations, action retrieval and classification can be solved effectively. Finally, we show that the activations and the fitting error alone are very useful for motion related disease diagnosis, thus demonstrating the intuitive nature of the description (Sec. 4.5.5).

We used three datasets, our own, the Univ. Bonn dataset, and the CMU MoCap dataset [Lab]. Our own dataset consists of 55 different actions, which were captured with the MOVEN motion capture suit [MOV] at 100 fps. Each action sequence consists of at least 6 repetitions of the same action. Fig. 4.3 shows some of the actions in our dataset.

The convergence speed primarily depends on OMP and Bregamn algorithm. OMP is a greedy algorithm that takes linear time, and Bregman is proved to be very efficient for many problems that are difficult by other means [GO09]. Thus, our algorithm is very efficient. It takes only 10-15 iterations before convergence ($\approx$5 mins in Matlab for our dataset).

4.5.1 Parameters

μ_1 and μ_2 in Eq. 4.4 determine the balance between the fidelity of the object function and the sparsity of the variables in the optimization. In all the experiments, they were both set to $\frac{1}{2}$. λ in the Split Bregman Algorithm (Eq. 4.12) is the parameter for penalizing the auxiliary variable. It was set to 1000 in our experiments. The length of the basis function was set to the frame rate of our MoCap suit (100). In our experiments, the normalization speeds up the convergence of the algorithm, therefore, MoCap sequences were also approximately normalized to 100 samples per action instance.

The initialization of Eq. 4.4 is randomly generated. This optimization is a large non-convex problem, and a common practice is to have a random guess at the beginning.

4.5.2 Learning the Basis Functions

Fig.4.3 visualizes 17 out of the 55 actions in our dataset. First, we applied our normalization algorithm. As found by visual inspection the discontinuities in the action sequences estimated by this algorithm correspond to the intuitive poses separating actions.

After normalization, the action decomposition algorithm is applied to the action sequences. Fig 4.5a shows the fifteen basis functions learned by the algorithm. Each column

is a color-coded basis function. In our modeling, individual joints are described by four basis functions, from the above set of fifteen learned basis functions. Fig 4.5b shows the basis functions used by the individual joints. We can see that different joints may share the same basis functions. Please note that the combination of the basis functions that composes joint movements. Different joints may have different combinations for an action, therefore, it is better to plot the functions individually.

Despite these very small numbers, the approximation is very good. The first column in Table 4.1 shows that the error residual in the approximation was very small. The residual was measured by the total fitting error divided by the number of frames and the number of joints in the dataset. On average, our representation approximates the training sequences with only 2.36 degree per joint in every frame.

The result shows that the primitives are effective and compact representations of the actions in the datasets.

Table 4.1: Average fitting error for different MoCap sequences using the basis functions learned in Sec. 4.5.2.

Seq	Training	Walking,Bonn	Running, Bonn
Error	2.36°	3.18°	3.56°

4.5.3 Motion Approximation and Compression

We use the basis functions to approximate novel actions. This further leads to effective compression of MoCap data. In this experiment, the novel sequences were first normalized, and Eq. 4.9 is then used to compute the activations and approximate the sequences. An averaging filter is used to handle the possible discontinuities between actions.

A useful representation of action should have the generalization capability of expressing unseen actions. First, we used the basis function learned from our dataset to approximate two sets of the sequences in the Bonn dataset, which were captured by an optical motion capture suit by different subjects. We then measured the fitting error. As shown in the 2^{nd} and 3^{nd} columns in Table 4.1, they are very small.

Comparing lossy compression results objectively is very difficult. As pointed out by [Ari06], the fitting error may not be a good predictor of visual quality. Therefore, the subjective judgments were used. Fig. 4.4b visualizes the approximation using two "salsa" dances in the CMU dataset. We can see that the poses of reconstructed movement (Fig.

4.4b) approximate those in the original sequences (Fig. 4.4b) very well. This side by side comparison shows that the shift-invariant decomposition effectively handles the complicated novel actions.

Another advantage of our decomposition approach is that it leads to high compression rate. To effectively compress MoCap sequences composed of arbitrary actions is very useful both for storage and for visualization, but it is also a challenging problem. In our approaches, a joint movement that has 100 data samples can be described by only four coefficients. Thus, we achieved approximately $25:1$ compression rate by default[3]. Table 4.2 compares the compression rate of some algorithms on CMU dataset.

Table 4.2: Comparison of different MoCap data compression algorithms. (*): the compression rate without quantizing the weights. (**): the compression rate with weight quantization. The ratios of the other algorithms are copied from [Ari06]

Algorithm	Ours	Arikan	Wavelet	Zip
Ratio	19:1(*) 37:1(**)	30:1	6:1	1.4:1

We archived competitive compression rate compared to the state-of-the art algorithm. More importantly, the primitive-based compression is fundamentally invariant to the frame rate. A major difference between the our approach and previous approaches is that we explicitly model the human actions. Therefore, the change in frame rate only changes the number of samples in the basis functions, but not the activation positions in time.

4.5.4 Action Retrieval and Classification

In the following experiments, we demonstrate the usefulness of our description for action classification and retrieval. First, our preprocessing algorithm breaks the action sequences into action segments. Each segment is treated as one complete action. Then, we decompose every single action using Eq. 4.9 allowing for only 1 activation. Finally, action retrieval and classification can be solved effectively using only the magnitude of the activations as the weights.

We considered it more helpful to provide the intuition of the usefulness of the representation using simple Euclidean distance and a neireast neighborhood classifier. This demonstrates how much the representation contributes to the retrieval and classification,

[3]For complicated actions, we may use more activations to approximate the time series, based on the normalization ratio. In addition, a small amount of overhead is required (*e.g.*, storing the scaling factor and the basis functions).

without tuning parameters in a sophisticated classifier. Therefore, we chose to compare our representation to decomposition methods.

4.5.4.1 Segment-based Action Retrieval

We compared our algorithm with the Sparse Principal Component Analysis [DGJL07] algorithm and the Principal Component Analysis algorithm. For both algorithms the segments are normalized to have the same length.

Retrieval is evaluated on our dataset using the so-called Bullseye test [LJ07]. 6 segments per action sequence were selected[4]. A leave-one-out trial was performed for every segment. The retrieval rate is ratios of the correct hits in top 12 candidates for all trials.

The performance of the three algorithms is shown in Table 4.3. Our algorithms achieved higher accuracy (86.07%) in the Bullseye test. This indicates that the repeated action segments in an action sequence have similar representation. The result demonstrates that our decomposition algorithm has the power of finding the similar movements.

Table 4.3: Performance comparison (Bullseye) of action retrieval on the segments of our dataset. Three algorithms, namely Sparse PCA, PCA and our algorithm, were used in the comparison. The segments were normalized for Sparse PCA and PCA.

Algorithm	Ours	Sparse PCA	PCA
Accuracy	86.07%	82.64%	78.87%

4.5.4.2 Segment-based Action Classification

We classify actions performed by different subjects. Four actions ("walking", "marching", "running", and "salsa") from the CMU dataset, were used in the experiment.

We compared our algorithm with the Sparse Principal Component Analysis algorithm and the Principal Component Analysis algorithm. To demonstrate the usefulness of the weights, we chose a very simple k-nearest neighborhood (kNN, $k = 3$) classifier. For each partitioning algorithm, we randomly selected 50% of the estimated segments in each action category as the training samples, and used the remaining as the test samples. Figs. 4.6a-c show the confusion matrices of the classification using the coefficients obtained by our algorithm, the Sparse PCA and the PCA algorithm.

[4]For action sequences which had a larger number of segments, we randomly selected 6 segments

Results show that our representation gives the best classification performance. This demonstrate that our shift-invariant representation models the nature of the human action, and the sparse linear decomposition facilitates the performance of classification.

4.5.5 Motion Disorder Diagnosis

The activations are a natural measurement for describing body coordination. If the activations for different joints are not well aligned, the subject might have a problem in controlling her/his motions. Another measure is the approximation error.

We demonstrate the applicability of the basis functions in modeling the Parkinson motion disorder, which is characterized by degenerative muscle movements. The primary symptoms are the results of decreased coordination caused by insufficient control. This problem is highly difficult because the correct modeling for the coordination among different body parts is challenging.

In this experiment, we captured the MoCap data for four patients diagnosed with the PD disease and four healthy controls (Table. 4.4). Fig. 4.7 shows the scenario when the experiments were performed. Subjects were asked to perform a number of actions repeatedly.

For this application, we learned the basis functions and the activations for each subject individually. Three common actions, "Finger To Nose", "Catching a Tennis Ball", and "Bread Cutting", were recorded. Figs. 4.8a-c show the plot of the activation alignment score and the average approximation error. The alignment score between two sequences is the zero-mean standard deviation of the differences between corresponding elements. The activation alignment score is defined as the largest value of the pairwise alignment scores.

Fig. 4.8d shows the chart for classifying the patients. As can be seen the two measurements are sufficient to separate controls from patients. Referring to 4.8a-c, the data points are well separated.

The diagnosis shows that the activations in our decomposition approach characterize the underlying rhythm in the parts of the bodies. This suggests that our approach is well suited for further understanding the principles of coordinated actions.

Table 4.4: Parkinson disease patients' age information. The disease level is measured by the Hoehn and Yahr scale which ranges from 1-5 (shown in parentheses).

Controls	Patients
4 healthy subjects	63(2.5), 63(2.5), 60(2.5), 60(3)

4.6 Chapter Conclusion

This chapter presented an algorithm for finding basic primitives to represent human motion data. Body movements in MoCap sequences are decomposed into the shift-invariant basis functions and their activations. The decomposition is solved by alternately updating two large convex problems using L_1 minimization techniques. Experiments show that the compact representation is effective for motion approximation, MoCap data compression, action retrieval, and classification with application to disease diagnosis.

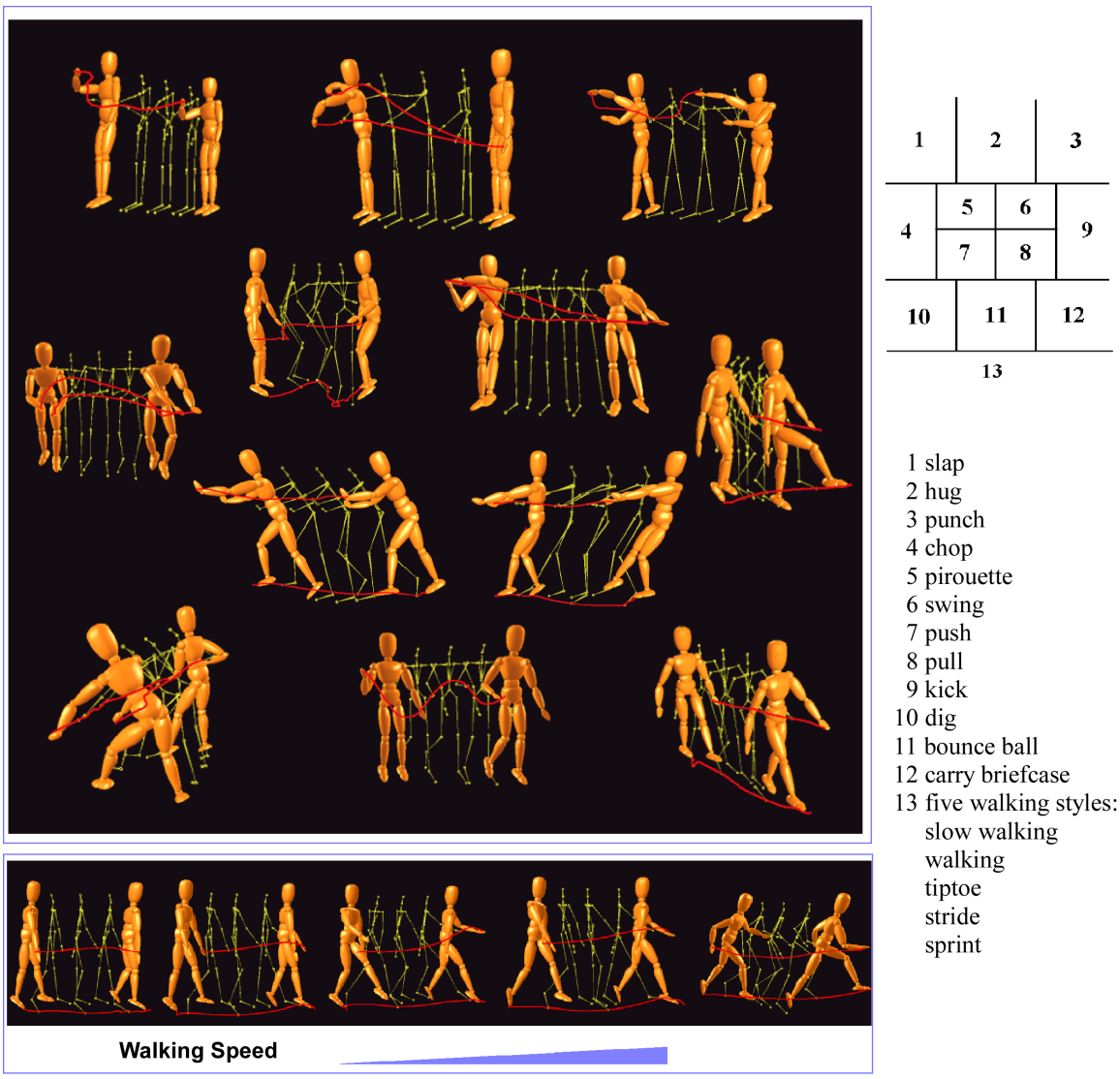
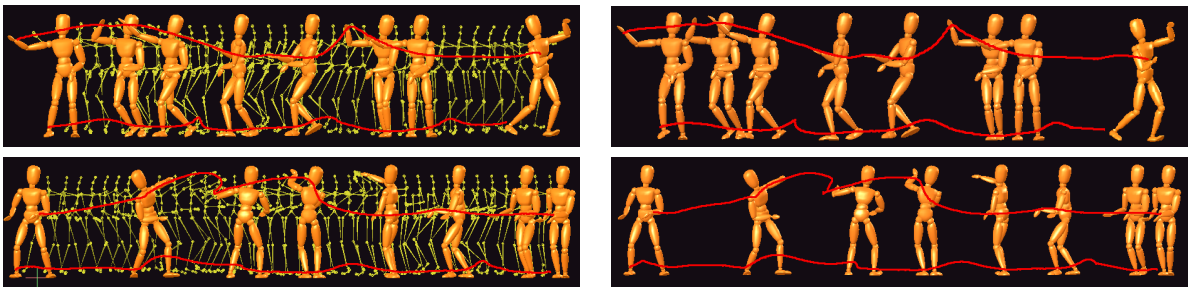


Figure 4.3: 17 out of the 55 actions in our data set. The rendering is as follows: poses corresponding to the discontinuities are displayed as mannequins; the transitions in between are illustrated by wire-frames; the trajectories of some joints are drawn in red.



(a) (b)

Figure 4.4: Side by side comparison between original motion frames (a) and reconstructed motion (b) using the estimated basis functions, demonstrated on two "salsa" sequences from the CMU dataset [Lab].

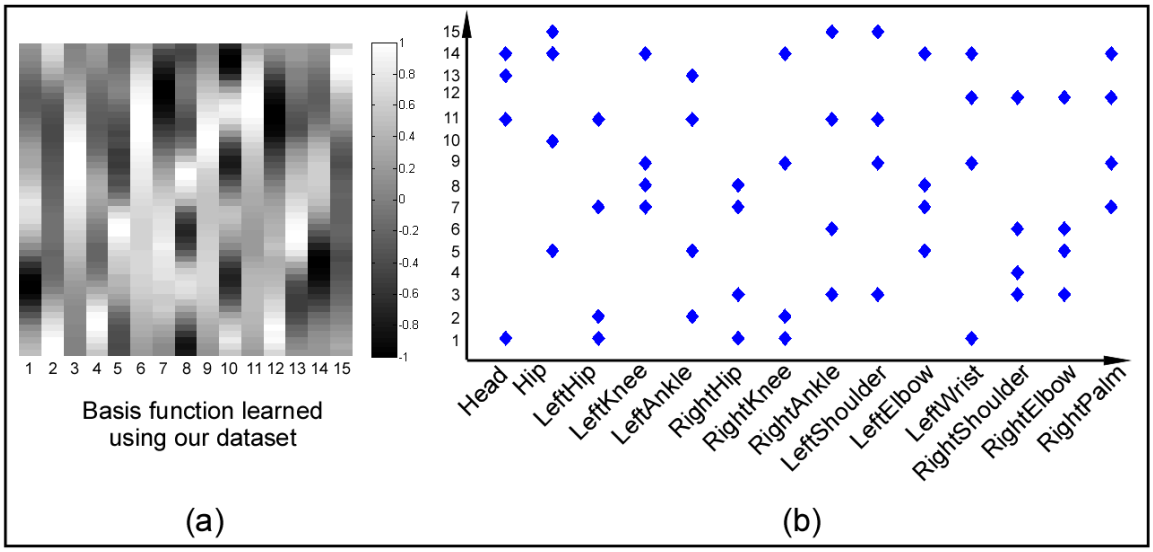


Figure 4.5: The basis functions learned by the algorithm (a) and their usage for individual joints (b) (denoted by the blue diamonds)

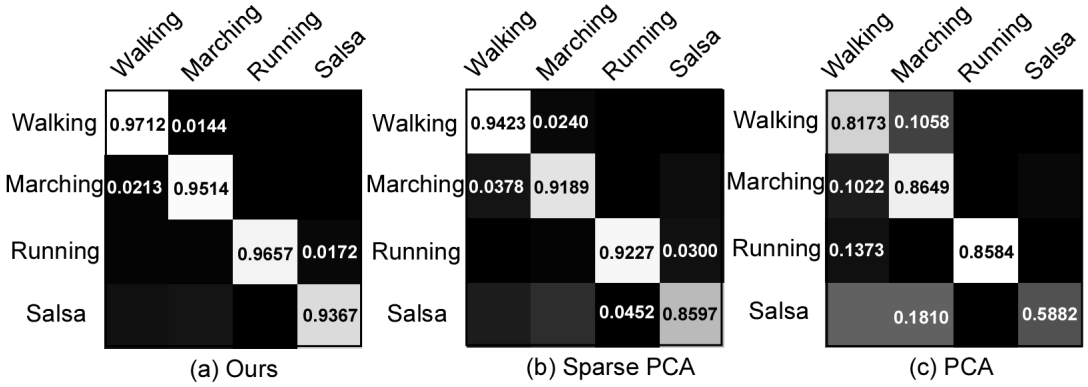


Figure 4.6: Action classification. Four actions, "walking", "marching", "running", and "salsa" from the CMU dataset, were used in the experiment. A very simple k-nearest neighborhood (kNN, $k = 3$) classifier was chosen. (a)-(c) show the confusion matrices of the classifier using the weights of the proposed algorithm, the Sparse PCA algorith, and the PCA, respectively. For each algorithm, 50% of the estimated segments in each action category were randomly selected as the training samples and the remaining as the test samples.

Figure 4.7: Collecting Parkinson Disease data. Courtesy of Leonardo Max Batista Claudino.

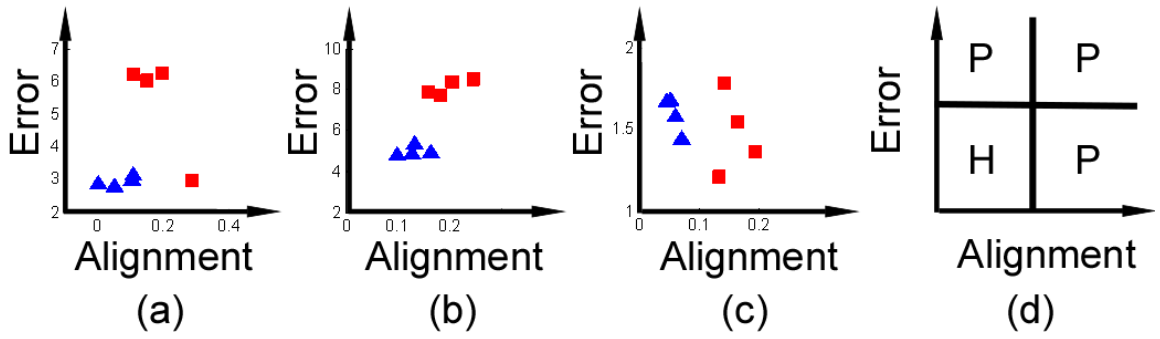


Figure 4.8: Parkinson disease diagnosis by measuring the alignment and the average approximation error. From left to right, the results for "Finger To Nose", "Catching a Tennis Ball", "Bread Cutting", and the diagnosis chart, respectively. In a)-c), blue triangles denote the healthy controls, and red squares denote the patients. d) suggests a chart for diagnosis. P: patient. H: healthy control.

Chapter 5

An Application of Granger Causality to Coordinated Actions Analysis in Orchestra

5.1 Chapter Summary

Causal knowledge is a concept well known in artificial intelligence and developmental learning. In many commonsense reasoning tasks, it is studied in a setting where the causes and the effects are discrete events. In this chapter we use the concept of causality to evaluate coordination of continuous human actions. The "coordination of actions" refers to the degree of synchrony or complementarity between actions performed by different individuals. We propose to use the Granger Causality for measuring causal interactions between human actions. In our approach, actions are modeled as autoregressive processes, and Granger Causality is framed in terms of predictability. The underlying idea is that if one action causes the other, then knowledge (history) of the first action should help predict future values of the latter. We carried out experiments using data of the trajectories of the instruments of eight orchestra musicians and the batons of two conductors. Our data demonstrates the existence of a dynamical network of causal interactions among players and conductors. The modulation of the network is in relation to the degree of direction that the conductor is able to express to the musicians. The study suggests that the concept of causality allows us to quantitatively study the coordination of actions, and also that the same idea may be applied to other human action scenarios.

This chapter is based on the paper accepted by the IEEE Conference on Developmental Learning 2010. I did not submit the final version of the paper because the co-authors would like to revise it and submit to a prestigious neuroscience journal.

5.2 Introduction

Causality, defined as "the relation between a cause and its effect, or between regularly correlated events or phenomena" by Merriam-Webster Dictionary, is one of the fundamental mechanisms in many commonsense reasoning tasks. The analysis of causality is also very helpful for modeling the functionalities of cognitive systems, especially with regards to efficient estimation of appropriate responses of causes and modification of an agent's behavior online.

The common conception is that causes always precede effects. Typically, causes and effects are regarded as discrete events. Therefore, most of the previous work in artificial intelligence and development was concerned with the causal links between discrete events. However, causality also exhibits itself in continuous signals.

In this chapter, we focus on causality in coordinated human actions. Coordination is one of the prerequisites for social interactions among intelligent agents. In fact, all animal species grouping for defensive, reproductive or hunting needs have evolved complex communicative behaviors to obtain coordinated actions [Fri08]. Recent studies further suggested that the "action mirroring circuit" might be tuned to action coordination rather than single action perception [FCO05].

The causality network plays an important role in supporting the coordination of actions [SHK09]. The coordination, at the individual level, can be modeled conceptually as a computation transforming visual information into motor control parameters. In such a context, movement kinematics of one individual must have statistical causal relation with the kinematics generated by another individual. Thus, the understanding of this underlying causality network will facilitate the research of many developmental learning problems, such as cognitive robotics, multiagent systems, and human-computer interaction.

Despite the progress in the study of causality, quantitative measurements of a dynamical causality network for coordinated signals remain elusive. The reason is that the formal analysis of causality networks requires an effective definition of causality between pairwise signals. For example, if we merely record the kinematics of two actions behaving randomly, we would always find some correlated parameters (*i.e.*, velocity). However, correlation does not necessarily imply causation. Therefore, a tool for formally studying the causality network of coordinated signals is necessary.

Originally from econometrics and neuroscience, Granger Causality [Gra69] is a statistical concept that provides a computationally feasible way for quantitatively describing the pairwise causal influence. Granger's proposal is that if a time-series y causes (or has an influence on) x, then knowledge of y should help predict future values of x. Thus, causality is framed in terms of predictability.

In our study, we examined the causality network of the coordinated musicians in a chamber orchestra using the kinematic data of their instruments. An orchestra is an interesting case because movement coordination is the main concern of the musicians and a successful communication is a pivotal feature for which they train for years.

A chamber orchestra is divided into multiple sections (*e.g.*, "Violin I" and "Violin II" in our study). Theoretically, players in the same orchestra section should have the same motion. However, each player might play slightly different according to his/her style and understanding. Our aim is to verify whether complex coordinated behaviors may be influenced by the causal interaction expressed by individuals.

We recorded the trajectories of the endpoints of the instruments used by eight violin players and two conductors (Fig. 5.1a). The players played fifteen pieces with a professional and an amateur conductor. In this experiment, our goal was to find out whether the expertise of the conductor is related to his/her influence on the musicians' performance, and if this in turn has an influence on the communication among the musicians.

Our results demonstrate the existence of a dynamical network of causal interactions among the players and the conductors. The strength of causal influence is related to the degree of influence that the conductor is able to express to the musicians. Thus, our study suggests that we may quantitatively express the causality of coordinate actions.

This chapter is organized as follows. First, we present the concept of Granger Causality as a statistical inference of the causal interaction between pairwise coordinated signals. Then, we show experiments and discuss the causality network. Finally, we review the literatures related to causality, coordination, and human motion modeling, and conclude the chapter.

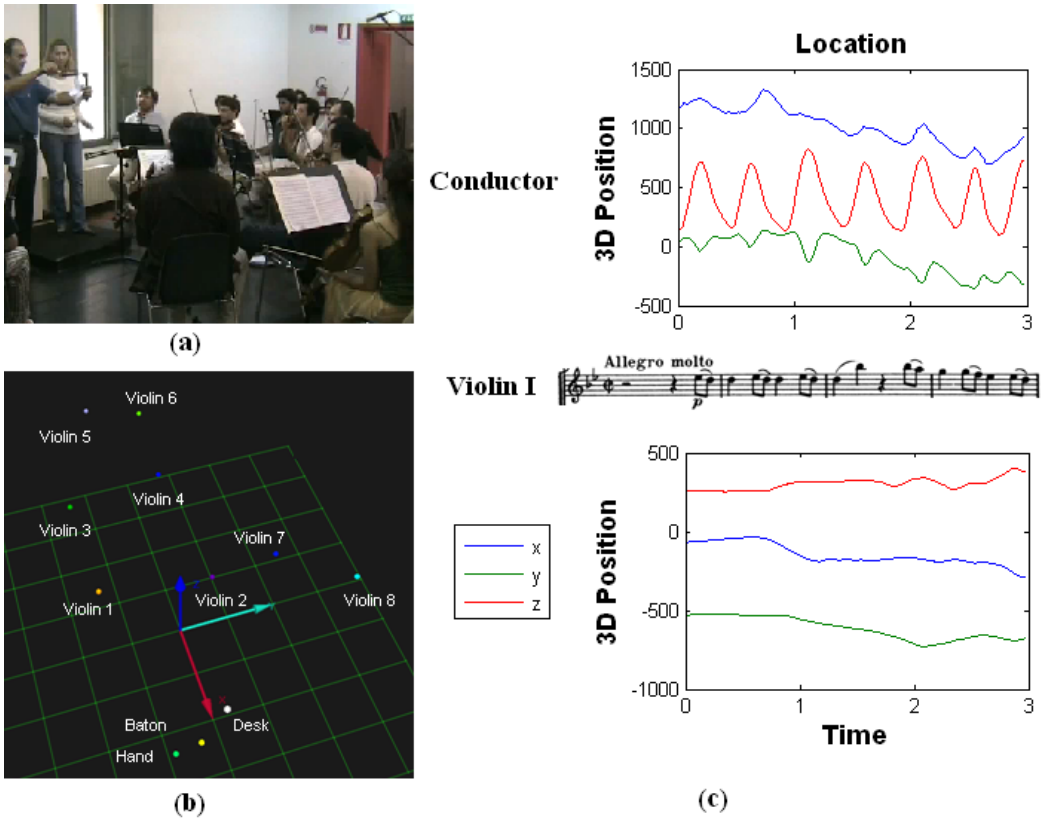


Figure 5.1: Illustrating the recording scenario and the kinematic data. (a) The subjects in our experiments. Eight string players (grouped into "Violin I" and "Violin II") were conducted by two conductors, respectively; (b) The locations of the endpoints of the music instruments; (c) The kinematic data of a conductor and a player in the first three seconds of the first movement of Mozart's Symphony No.40. The time series in red, green, and blue represent the trajectories of the markers in X, Y, and Z dimension, respectively.

5.3 A Brief Introduction to Granger Causality

The Granger Causality Test is a technique for determining whether one time series is useful in forecasting another. First, we discuss how to model time series using Autoregressive (AR) Processes. Then, we describe the idea of Granger Causality, which uses linear predictors.

5.3.1 Preliminary: Forecasting Time Series using Autoregressive Processes

Granger Causality is based on the residual in forecasting time series using linear predictors. An autoregressive model or process is a stochastic process. An $AR(k)$ process, where

k is the order of the model, has the following form:

$$y(t) = \sum_{i=1}^{k} a(i)y(t-i) + e(t), \tag{5.1}$$

where $a(i)$, $i = 1...k$, are the coefficients for the lagged observations, and e is the estimation residual.

Two criteria are widely used for selecting the optimal order of a linear predictor in Eq. 5.1: Akaike's Final Prediction Error (AIC) criterion and Schwarz's Bayesian Criterion (BIC) [BJ90]. Lutkepohl [Lut07] compared these and other order selection criteria in a simulation study and found that BIC chose the correct model order most often and led, on the average, to the smallest mean-squared prediction error of the linear models.

In many real world tasks, the time series is multi-dimensional. Human motion, for instance, is characterized by a number of continuous signals, such as a $3d$ trajectory of an instrument's endpoint in our study. Certainly we could consider each dimension as a $1d$ time series and use Eq.5.1 to model the data. However, these $1d$ time series could be highly correlated (*e.g.*, they might accelerate at the same time). Therefore, it is better to jointly model them.

The Vector Autoregressive model [NS01] is a natural extension of the traditional AR model. This predictor models a vector series as follows:

$$Y(t) = \sum_{i=1}^{k} A(i)Y(t-i) + E(t), \tag{5.2}$$

where $Y(t)$ is the d (dimensions) by 1 vector, E is the d by 1 residual vector, and $A(i)$, $i = 1...k$, is the d by d coefficient matrices, which can be learned using maximal likelihood estimation or the Bayesian prior (Bayesian Vector Autoregressive Process) [LP09].

5.3.2 Granger Causality Test

Given two time series X_1 and X_2, we may either model them separately using two individual AR processes (denoted as L_1), or jointly model them in one AR process (denoted as L_2).

If X_1 and X_2 are modeled separately, we have:

$$X_1(t) = \sum_{i=1}^{k} A_1(i)X_1(t-i) + E_1(t) \tag{5.3}$$

$$X_2(t) = \sum_{i=1}^{k} A_2(i)X_2(t-i) + E_2(t), \tag{5.4}$$

where A_1 and A_2 are the coefficients for the lagged observations, and E_1 and E_2 are the residuals.

Alternatively, we can use a more complicated model L_2 to jointly model these two signals as follows:

$$X_1(t) = \sum_{i=1}^{k} B_{11}(i)X_1(t-i) + \sum_{i=1}^{k} B_{12}(i)X_2(t-i) + \overline{E}_1(t) \tag{5.5}$$

$$X_2(t) = \sum_{i=1}^{k} B_{21}(i)X_1(t-i) + \sum_{i=1}^{k} B_{22}(i)X_2(t-i) + \overline{E}_2(t), \tag{5.6}$$

where B_{11}, B_{12}, B_{21} and B_{22} are the coefficients for the lagged observations, and $\overline{E}_1$ and $\overline{E}_2$ are the residuals.

The Granger Causality Test focuses on the improvement of the residual. Denote

$$e_i = \sum E_i(t)^2 \tag{5.7}$$

$$\overline{e}_i = \sum \overline{E}_i(t)^2 \tag{5.8}$$

for i=1,2. Since L_2 involves more parameters than L_1 and has more degrees of freedom, necessarily L_2 improves the performance of prediction ($\overline{e}_i < e_i$). However, the improvement could be due to the knowledge (history values in the time series) of the other signal, or it could be simply due to the increase of the degree of freedom in the model. Therefore, F-Test is used to validate this hypothesis.

F-Test is a classical approach to compare statistical models for data fitting, and further to identify the model that best fits the population from which the data was sampled.

Denote

$$\mathcal{F}_{2\to1} = \frac{\frac{e_1-\overline{e}_1}{p_2-p_1}}{\frac{\overline{e}_1}{n-p_2}} = \frac{e_1-\overline{e}_1}{\overline{e}_1}\frac{n-p_2}{p_2-p_1}, \tag{5.9}$$

and

$$\mathcal{F}_{1\to 2} = \frac{\frac{e_2-\overline{e}_2}{p_2-p_1}}{\frac{\overline{e}_2}{n-p_2}} = \frac{e_2-\overline{e}_2}{\overline{e}_2}\frac{n-p_2}{p_2-p_1}, \tag{5.10}$$

where p_1 and p_2 are the degrees of freedom in L_1 and L_2, respectively, and n is the total number of observations in the input signal. In our study, $p_1 = 2kd^2$ and $p_2 = 4kd^2$.

In the F-Test, $\mathcal{F}_{2\to 1}$ and $\mathcal{F}_{1\to 2}$ are assumed to have F-distribution with $(p_2 - p_1, n - p_2)$ and $(p_2 - p_1, n - p_1)$ degrees of freedom, respectively. $\mathcal{F}_{2\to 1}$ ($\mathcal{F}_{1\to 2}$) denotes the improvement of predicting X_1 (X_2) using the information from the history of X_2 (X_1). There is no simple arithmetic relationship between $\mathcal{F}_{2\to 1}$ and $\mathcal{F}_{1\to 2}$. We consider $\mathcal{F}_{2\to 1}$ as the "causal influence" that X_2 passes to X_1, and similarly we call $\mathcal{F}_{2\to 1}$ the cause from X_2 to X_1.

The Granger Causality Test claims that X_1 "Granger causes" X_2 if and only if

$$\mathcal{F}_{1\to 2} > \mathcal{F}_{2\to 1}. \tag{5.11}$$

For the orchestra performance considered here, we may interpret this result as the causal influence from X_1 to X_2 being larger than the one from X_2 to X_1 such that X_1 "drives" X_2.

5.3.3 Statistical Analysis of Granger Causality

In many real world scenarios, signals are non-linear, and linear models are regarded as local approximations of the non-linear signals. Thus, the time series is divided into multiple windows, and linear analysis is carried out in each window.

To analyze the overall causal influence between two non-linear time series, we compute the Granger Causality for each window, and use kernel density estimation [Par62] to summarize the causal influences between the signals. Kernel density estimation is a non-parametric method for estimating the probability density function of a random variable. Isotropic kernels are frequently used if no domain-specific prior is given [DHS00]. This provides a more robust measurement of the expected value of the variable in many studies.

5.4 Experiments

In this section, we present our experiments and suggest how Granger Causality can be used for studying coordinated human actions. We demonstrate a causality network that is

modulated by the influences among coordinated actions of the members of an orchestra. This network can be conceived as a conversation between several individuals.

First, we present the data acquisition and preprocessing procedures. Second, we show the results of the experiments. Specifically, we show that a professional conductor has a greater influence than an amateur on average, and influences the orchestra more frequently. Furthermore, we show that the musicians in the same orchestra section tend to be less reliant on the influence of peers when the influence from the conductor is greater. Finally, we compare our results to the non-causal data obtained by scrambling the causal data, and demonstrate the significance of the above results.

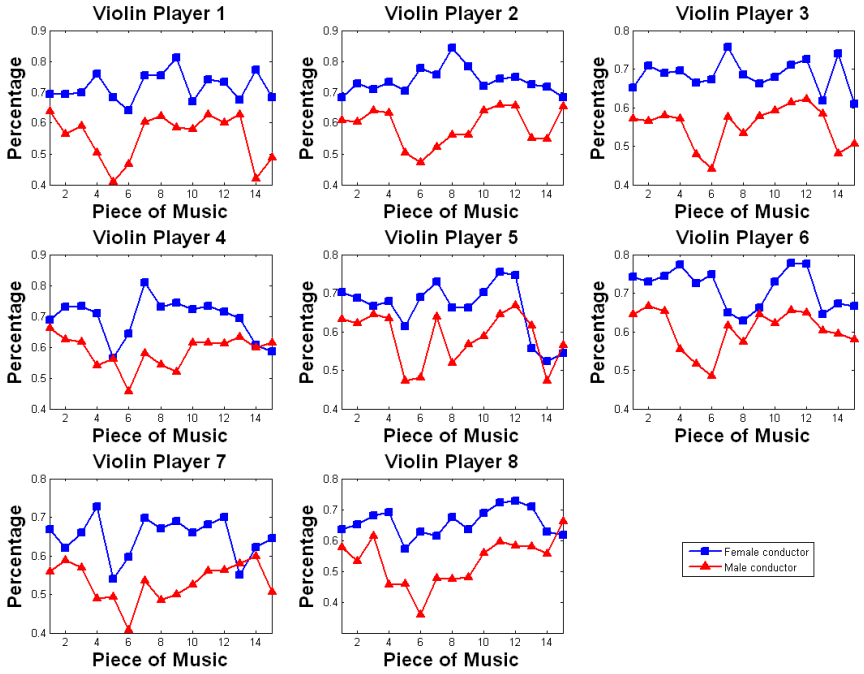


Figure 5.2: The percentage of the time when the conductor leads the players. This figure shows that the professional conductor consistently leads the players more frequently than the amateur conductor.

5.4.1 Data Acquisition and Preprocessing

5.4.1.1 Data Acquisition

We recorded the absolute positions of the endpoints of the music instruments. Data acquisition was performed using a Qualisys system consisting of three cameras recording the

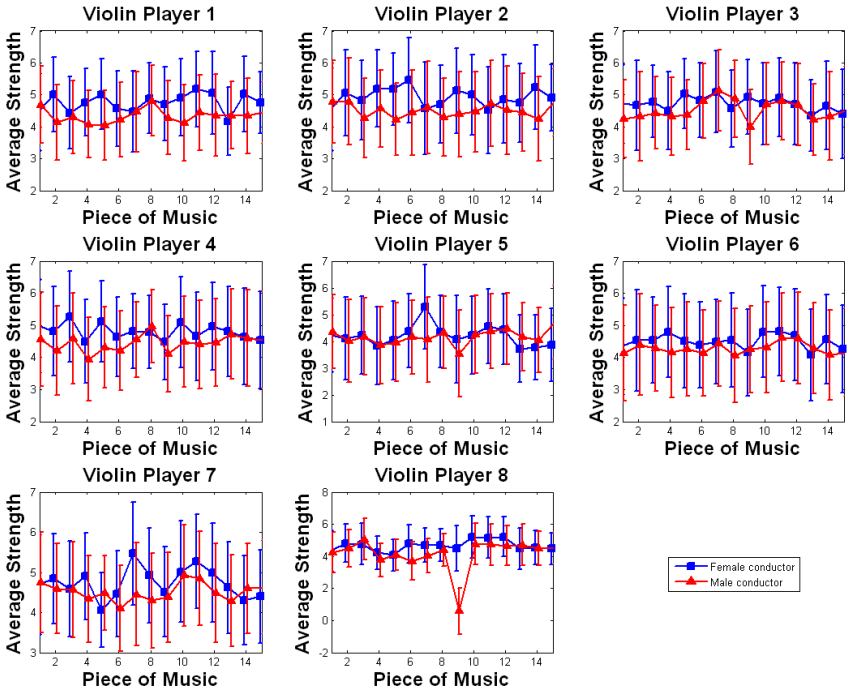


Figure 5.3: The error bar of $\mathcal{F}_{c\to i}$, which denotes the causal influence strength from the conductor to the players. This figure shows that the professional conductor's influence is consistently stronger than the amateur.

positions of the markers on players' bows and conductors' batons at 240Hz. Fig. 5.1b illustrates the positions of the instrument endpoints in the environment and Fig. 5.1c shows two fragments of the recordings.

5.4.1.2 Data Preprocessing

We used the spline method [Deb78] to handle missing data in the $3d$ trajectories and low-passed the signal for artifact removal. The data was then divided into overlapping windows. Two consecutive windows have $\frac{1}{3}$ overlap . The window size is set to 2 seconds to incorporate a reasonable amount of information.

We select the order of the AR processes as follows. First, we computed the AR model order in each window using the BIC criterion. Then, we used kernel density estimation to compute the expected value of the order and set $k = 8$.

The AR model assumes that the input time series is stationary, which means the eigenvector of the model must be inside the unit circle [Hay]. If this does not hold, one com-

mon practice is to differentiate the signal because the derivative may provide better results practically. In our experiment, we found that the second order derivative of the $3d$ motion series satisfies this assumption. Essentially, we used the acceleration of the kinematic data as the input.

The use of higher order derivatives may seem to make the system more sensitive to noise. In our experiment, we found that the acceleration is reasonably accurate, and we use a Gaussian filter to remove the noise.

The usefulness of the acceleration further shows the importance of the causality in the physical interaction. This may be very important to understanding how a human being cause or change the behavior of others by modifying their planned actions.

5.4.1.3 Creating Non-Causal Data for Comparison

In order to test the statistical significance of our estimates, we need to verify the Granger Causality result against the null hypothesis of non-causal relations. In other words, the Granger Causality Test may indicate causal interactions between two random actions that do not have any causality, therefore, the causal relationship is valid only when the value is larger than a statistical threshold obtained from non-causal data.

Intuitively, the movements of two players at different times should have no causal interaction statistically. Therefore, we randomized the windows for each musician to create the non-causal data for comparison in our experiments.

5.4.2 Experimental Results

In the experiments, we numerically quantified how the musicians accommodate their performance according to the causal information passed by other musicians in a causality network. The idea is that different conductors exhibit different influences towards musicians. This difference, in turn, also affects the causal influences among players.

First, we computed the causality flow between the conductors and the players. Then, we computed the average influence of causality within the same orchestra sections under the direction of different conductors. Finally, we compared the above results to the outcomes from the non-causal data, and to demonstrate that the results in the previous experiments are significantly meaningful.

5.4.2.1 Experiment 1: The Causality between the Conductors and the Players

This experiment shows that the professional conductor influenced the orchestra more frequently than the amateur conductor, with a stronger average causal interaction.

Denote

$$D_i(m) = \begin{cases} 1, & \text{if } \mathcal{F}_{c\to i}(m) > \mathcal{F}_{i\to c}(m) \\ 0, & \text{otherwise} \end{cases} \tag{5.12}$$

where $\mathcal{F}_{c\to i}(m)$ denotes the causality from the conductor to the i^{th} player in the m^{th} window. Fig. 5.2 shows that the percentage of the time (*i.e.*, the expected value of $D_i(m)$) when the conductor leads the players for the fifteen pieces of music and the eight players.

This experiment shows that the professional conductor has a higher percentage than the amateur conductor in 95.8% of the cases. Interestingly, Fig. 5.2 also shows that in some pieces the amateur conductor has less than 50% "leading time" for some players (*e.g.*, Player 1 in the 5^{th} piece), which means the players actually "lead" the conductor.

We further show that the professional conductor has a stronger influence than the amateur conductor on average. We computed the average causality flow from the conductor to the players for each piece of music (Fig. 5.3). The error bar visualizes that in 77.51% of the time the professional conductor has a stronger causal influence than the amateur.

The analysis demonstrates that a successful coordination is due to the effective causal information passing between individuals. The causality, on one hand, helps the agent understand the peer's intentions based on contextual information. On the other hand, it enables efficient prediction, anticipation and planning of appropriate actions in response.

5.4.2.2 Experiment 2: The Causality within the Players

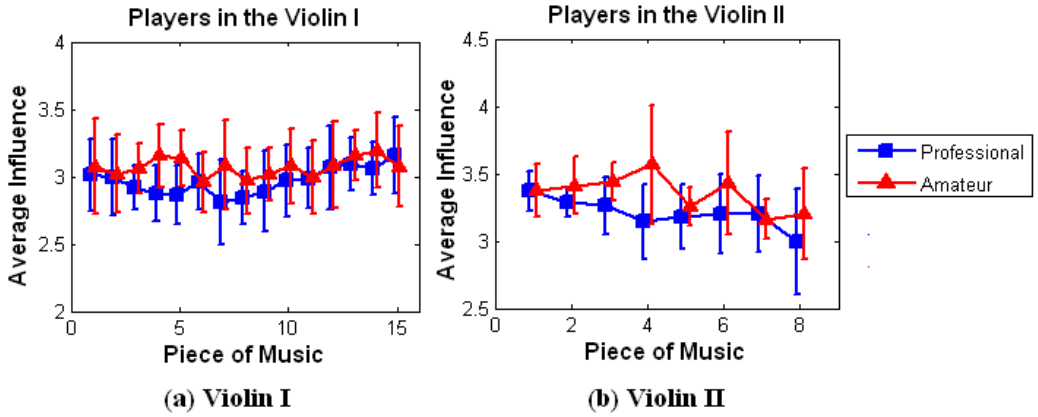
This experiment demonstrates that the players in the same orchestra section (*i.e.*, "Violin I" and "Violin II") have different behaviors in terms of causal interactions when they play under the direction of two conductors (Fig. 5.4).

We represent the level of coordination for each player using the maximal value of the causal interactions from his/her peers in the m^{th} window as follows:

$$\mathcal{F}_i(m) = \max_j \mathcal{F}_{j\to i}(m), \tag{5.13}$$

where i and j are the players in the same orchestra section.

Denote $\mathcal{F}(m)$ as the expected value of $\mathcal{F}_i(m)$ for all the players in the same orchestra section. It can be regarded as the average "in-group" influence in the m^{th} time window.



(a) Violin I (b) Violin II

Figure 5.4: The average influence and its standard deviation of the causality among the players in the same orchestra section. (a) The players in the "Violin I"; (b) The players in the "Violin II" (The 9^{th}-15^{th} piece of music are not shown due to many missing samples).

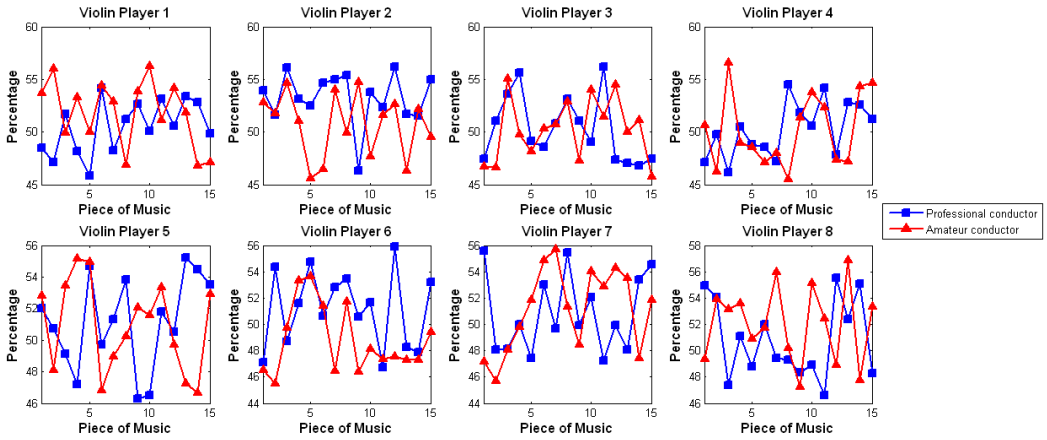


Figure 5.5: The percentage of the time when the conductor leads the players in the non-causal data.

Fig. 5.4 shows the average in-group influence in these two orchestra sections for each piece of music. Under the direction of the professional conductor (whose influence is larger) players have less influence on each players. The "Violin I" and the "Violin II" have weaker average in-group causal interaction with the professional conductor than the amateur, with 6.18% and 6.04% relatively on average, respectively.

This experiment demonstrates that a dynamical causality network supports the coordination of the musicians. Each player has to follow the fundamental direction of the

conductor and accommodate his/her performance of the peers. If the conductor is unable to convey enough causality, the players adjust their internal communication by increasing the necessary causal interactions among themselves.

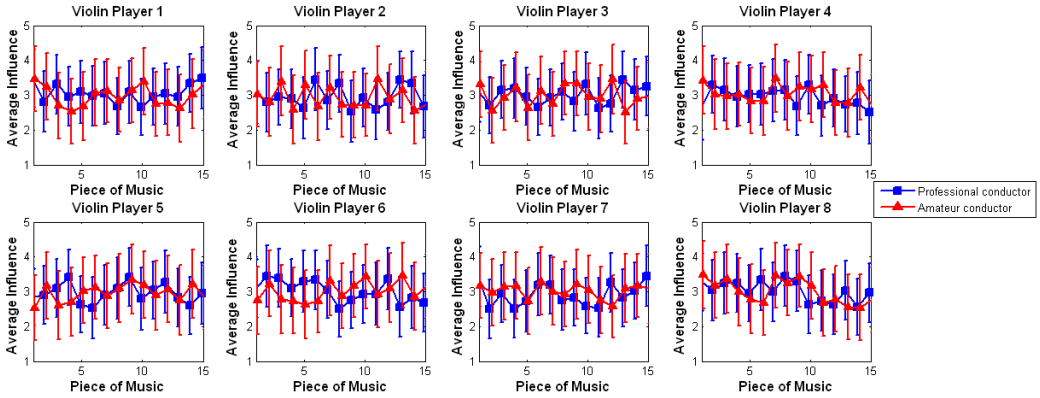


Figure 5.6: The error bar of $\mathcal{F}_{c\to i}$, which denotes the causal influence from the conductor to the i^{th} player, in the non-causal data.

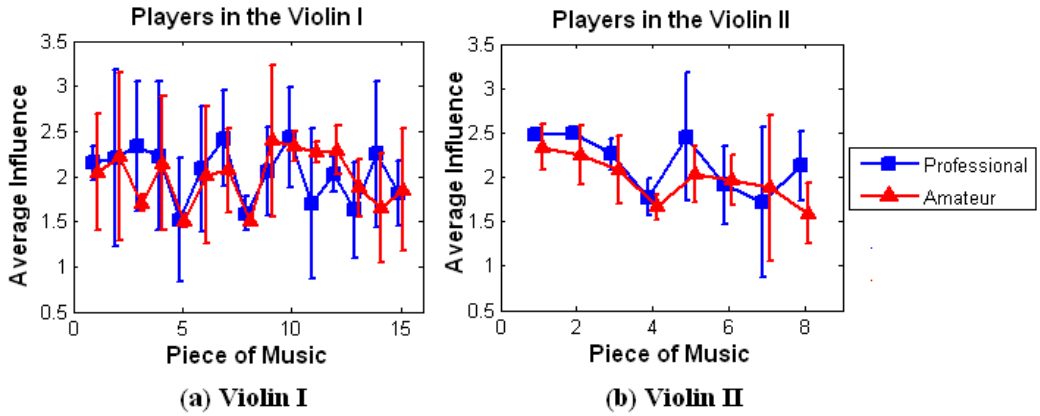
5.4.2.3 Comparison using the Non-Causal data

We repeated the above experiments using non-causal data obtained by randomizing the trajectory windows. The same procedures in the previous experiments were used to test the statistical significance of the results.

Figs. 5.5 and 5.6 show the results for the non-causal data. Compared to Experiment 1, the percentage of leading time (Fig. 5.5) is approximately close to 50% for both conductors, and the average influence (Fig. 5.6) is 3.04 (37.1% smaller than Fig. 5.3). These figures demonstrate that there is no significant difference between the two conductors in the randomized data. Thus, this shows that the results in Experiment 1 are useful for assessing the causality network.

Fig. 5.7 shows that the average in-group influences in non-causal data are 1.92 and 2.01 for the professional conductor and the amateur, respectively. Compared to Fig. 5.4, one can see the in-group influence in causal data is 31% larger. Fig. 5.7 also does not suggest a difference in coordination between the players under two conductors. The results show that the causal interactions within the players are significant, and the interactions contribute to the coordination of actions. This further demonstrates the importance of a dynamical causality network in coordinated actions of music performance.

These two comparisons suggest that the results of Experiment 1 and 2 are valid and meaningful. The experiments show the usefulness of Granger Causality in analyzing coordinated actions. The difference between the two conductors is due to the causal information passing between individuals in a dynamical causality network.



(a) Violin I (b) Violin II

Figure 5.7: The average strength and its standard deviation of the causality among the players in the same orchestra section using the non-causal data. (a) The players in the "Violin I"; (b) The players in the "Violin II".

5.5 Chapter Conclusion

The aim of this study was to quantitatively demonstrate the existence of a dynamical network of causal interactions among coordinated actions. Specifically, we examined the Granger Causality between the conductors and the players (Experiment 1), and the in-group causal interaction among the players under different conductors (Experiment 2). The study suggests that causality is a useful concept for analyzing coordinated actions, and that it may be applied to many human interaction scenarios.

Chapter 6

Detecting Discontinuity in Human Movement for Action Representation and Recognition

6.1 Chapter Summary

We propose an approach for detecting discontinuities of human movements in videos and motion capture data. Our approach aims at providing natural temporal segmentations of human actions, which can serve as a tool for action analysis and recognition. The idea is based on the jerks (third order derivatives) of the signals. For videos, we first reduce the dimension of the visual space using Gaussian Process Dynamical Models (GPDM). Then the discontinuities are estimated by finding the clusters of maxima of the jerks in the embedded signals. We demonstrate by examples that the action discontinuities in videos consistently correspond to the changes in the underlying body dynamics. Experiments suggest that the approach is robust to changes in viewpoint and background, and that the proposed algorithm finds consistent poses, which lead to better recognition rates.

This chapter is based on the paper appear in the First Workshop of Social Signal Processing in conjunction with the IEEE Conference on Affective Computing 2009. A video lecture is available at:

`http://sspnet.eu/2010/08/ieee-ssp-workshop-september-2009/`.

6.2 Introduction

Many recent studies in computer vision have been concerned with recognizing human activity. Ideally, we would like a formalism that allows us to describe actions in a way that facilitates recognition, characterization, and visualization. Although it is not clear yet how this formalism should be, it is very clear that whatever the approach may be, we need to be able to break the human movements into meaningful simpler, shorter movements.

How can we find these break points? If we have muscle signals over time, it is natural to partition the data at the points where we have discontinuities in the acceleration [ddSB03], an empirical observation supported by Newton's law. The poses corresponding to the discontinuities become natural segmentations of human movements, which in turn facilitate human action analysis.

In this chapter, we present an intuitive and efficient algorithm for detecting the discontinuities in unrestricted human movement both in videos and in motion capture (MoCap) data. We empirically found that the segmentations in videos and in MoCap data are consistent with each other. The reason for the consistency probably is because groups of muscles are exercised consistently, as has been found in [ddSB03] and are termed the "muscle synergies". The estimated discontinuities provide effective measurements for human action analysis in videos that are robust under different conditions including subject variation, video quality, frame rate, camera motion, and changes in viewpoint.

In motoric space, we find that different body joints are activated coherently. The activations of the different joints are estimated as the maxima of the changes in acceleration of individual joint angles. An action discontinuity occurs when a number of joints are activated at nearly the same time. Based on this fact, we propose an alignment algorithm to robustly detect the clusters of jerk maxima of different joints over time.

In visual space, certainly, we cannot measure the muscle signals in videos from any view. However, we can empirically transfer the idea of discontinuities to the visual domain using dimension reduction. We detect the discontinuities in reduced visual space obtained by the Gaussian Process Dynamical Model (GPDM). This model has been proposed recently for analyzing human MoCap data, and we extend it here to videos of single subjects moving in uncluttered scenes. We found that when there is an action discontinuity in the video, the changes in acceleration occur simultaneously along the different GPDM

dimensions. This allows us to use the same alignment algorithm for detecting action discontinuities in videos as well as in MoCap data.

Our key contributions are:

1. We propose an algorithm for creating natural temporal segments of human actions in videos and in MoCap data, based on the concept of action discontinuities.

2. We demonstrate that the action segments in visual space and in motoric space are consistent.

3. We show that the poses corresponding to the action discontinuities effectively improve the action recognition rates on public datasets.

The remainder of the chapter is organized as follows. Sec. 6.3.1 and 6.3.2 present the algorithm for detecting action discontinuities in MoCap data and in videos. Sec. 6.4 shows experiments and comparisons, and Sec. 6.5 concludes the chapter with a discussion .

6.3 Action Discontinuities in Human Movement

This section presents the discontinuity detection algorithm for human movements. In Sec. 6.3.1, we analyze MoCap data. We directly compute the change in acceleration in the motion capture data, and propose an alignment algorithm for finding the clusters of changes across different joints. In Sec. 6.3.2, we use dimension reduction algorithm to map videos to a low dimensional space. On this data we then detect the discontinuities using the same alignment algorithm as for MoCap data.

6.3.1 Action Discontinuities in Motion Capture Data

The data from a motion capture suit are time series of three rotation angles each at a number of joints on the human body. The quaternion series of a certain joint is a 4D vector

$$\mathbf{X}(t) = [x_1(t), x_2(t), x_3(t), x_4(t)]^T \tag{6.1}$$

The jerk of $\mathbf{X}(t)$ is computed as

$$J(t) = ||\frac{d^3(\mathbf{X}(t))}{dt^3}||, \tag{6.2}$$

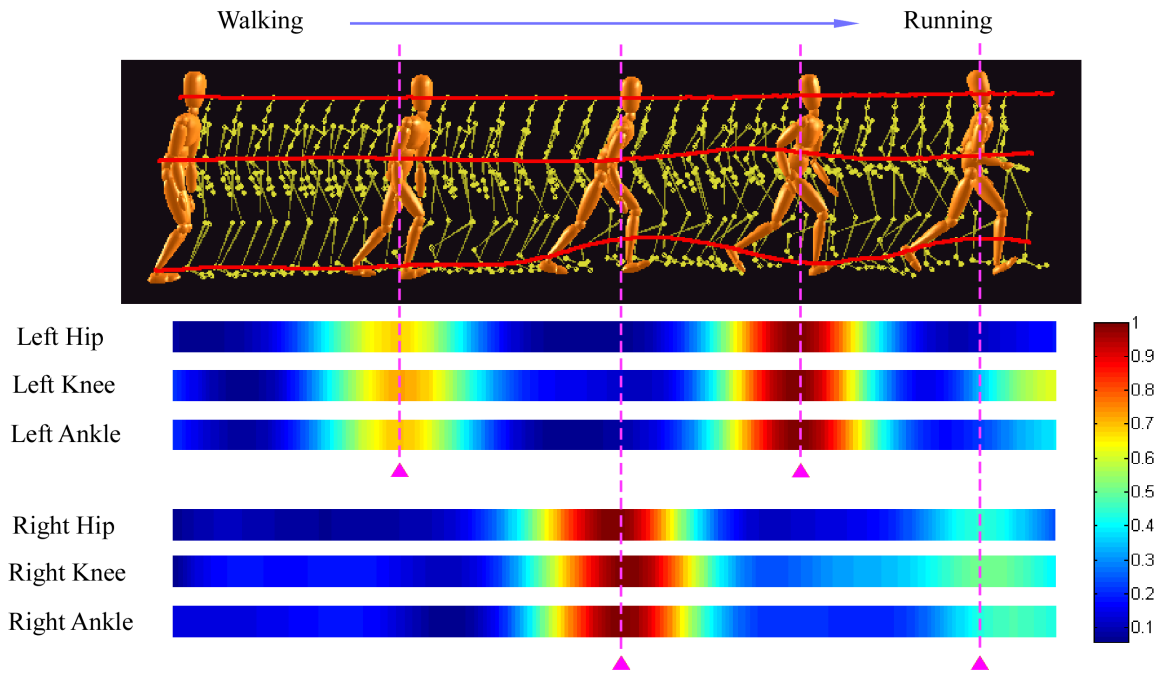


Figure 6.1: I redraw Fig. 4.2 here for convenient purpose. Action discontinuities of the speed-varying action "walking to running" from the University of Bonn dataset [MRC+07]. The jerk envelopes of the six joints are color coded and normalized in magnitude. The purple dashed lines indicate that the locations of the envelope extrema of certain joints coherently occur at the same time. The trajectories of the head, the left elbow, and the right ankle are drawn in red.

where $||\cdot||$ is the L_2 norm.

We need a robust measurement to estimate the jerk, because the estimation of higher order derivatives may be sensitive to noise. We thus compute the jerk envelope as

$$\text{Env}(t) = ||\text{Hilbert}(J(t))|| \tag{6.3}$$

for every joint, where Hilbert$(\cdot)$ is the Hilbert transform. This is a standard approach for computing the signal envelope [Bra99]. Then we process Hilbert$(\cdot)$ using a low pass Butterworth filter, and compute the extrema of the filtered signal Env(t) for every joint.

An action discontinuity occurs when there are a number of joints activated nearly at the same time. This is demonstrated in Fig. 6.1 for the speed-varying action "walking to running" from the University of Bonn dataset [MRC+07]. Here, we show the jerk envelopes of the joints "Hip", "Knee", and "Ankle" of both legs. The envelopes are color coded and normalized in magnitude. Since not all joints need to be activated together in an action,

our goal is to robustly detect the temporal clusters of the activated joints.

We model this problem by finding the optimal "alignment" of the envelope extrema for the different joints (the purple dashed lines in Fig. 6.1). We define the optimal alignment as the collection of the envelope extrema of the different joints with the minimal sum of the pairwise distances. The goal is to choose one of the discontinuities from each latent variable such that the sum of the pairwise distances is minimal.

The alignment algorithm automatically detects the activated joints over time. This clustering method is applicable to many scenarios where other measurements, such as global curvature, may fail. Alternative alignment approaches, such as dynamic programming, focus on global alignment and may align discontinuities at completely different locations.

We use an indicator function δ_m^i for the location of the i^{th} discontinuity of the m^{th} latent variable I_m^i such that

$$\delta_m^i = \begin{cases} 1, & \text{if } I_m^i \text{ is selected} \\ 0, & \text{otherwise} \end{cases} \tag{6.4}$$

for all possible i, and

$$\sum_i \delta_m^i = 1, \tag{6.5}$$

for $m = [1, ..., 4]$.

The distance between the locations of two discontinuities I_m^i and I_n^j $(m \neq n)$ is defined as

$$d(I_m^i, I_n^j) = \exp(\frac{1}{\sigma}|I_m^i - I_n^j|) - 1, \tag{6.6}$$

where σ is the parameter for the distance measurement.

Then, the cost of the consistency between a set of discontinuities in different latent variables is the sum of the pairwise distances,

$$c = \sum_m \sum_{n,n\neq m} \sum_i \sum_j \delta_m^i \delta_n^j d(I_m^i, I_n^j). \tag{6.7}$$

The optimization problem is therefore to find δ_m^i which minimize c,

$$\delta^* = \arg\min_\delta c \tag{6.8}$$

subject to the constraints stated in Eq. 6.5.

Eq. 6.8 is a constrained quadratic integer programming optimization problem. This can be reformulated to an unconstrained quadratic programming problem by relaxation and changing variables.

δ_m^i is relaxed to

$$0 \leq \delta_m^i \leq 1, \tag{6.9}$$

and the variables are changed to

$$\delta_m^i = \frac{\exp \theta_m^i}{\sum_i \exp \theta_m^i}, \tag{6.10}$$

where

$$-\infty \leq \theta_m^i \leq \infty. \tag{6.11}$$

Therefore, if we pre-compute the cost matrix M where $M(m, n, i, j) = d(I_m^i, I_n^j)$, we simply need to find the θ_m^i that minimize

$$C = \sum_m \sum_{n, n \neq m} \sum_i \sum_j \frac{\exp \theta_m^i}{\sum_i \exp \theta_m^i} \frac{\exp \theta_n^j}{\sum_j \exp \theta_n^j} M(m, n, i, j). \tag{6.12}$$

After minimizing Eq. 6.12 with respect to θ_m^i, $\delta_m^i = 1$ if and only if the value of $\exp \theta_m^i / \sum_i \exp \theta_m^i$ is maximal for all possible i in the m^{th} latent variable, and $\delta_m^i = 0$ otherwise.

The number of variables in Eq. 6.12 could be large (e.g., 100-500). An efficient solver for this quadratic optimization is necessary. We use L-BFGS in the proposed approach[1]. The alignment algorithm is iteratively applied until the value of C is smaller than a threshold ϵ.

6.3.2 Action Discontinuities in Videos

A human motion video is represented by the variables of a Gaussian Process Dynamical Model (GPDM) in a low dimensional space. The discontinuities are characterized by the maxima of the third order derivative of the latent variables, and the same alignment algorithm in Sec 6.3.1 is used for locating the discontinuities.

Gaussian Processes have become popular for regression and classification in machine

[1] www.cs.ubc.ca/~schmidtm/Software/minFunc.html

learning. A Gaussian Process is a collection of random variables which have joint Gaussian distributions [RW05]. It can be regarded as a probability distribution over functions in Bayesian inference.

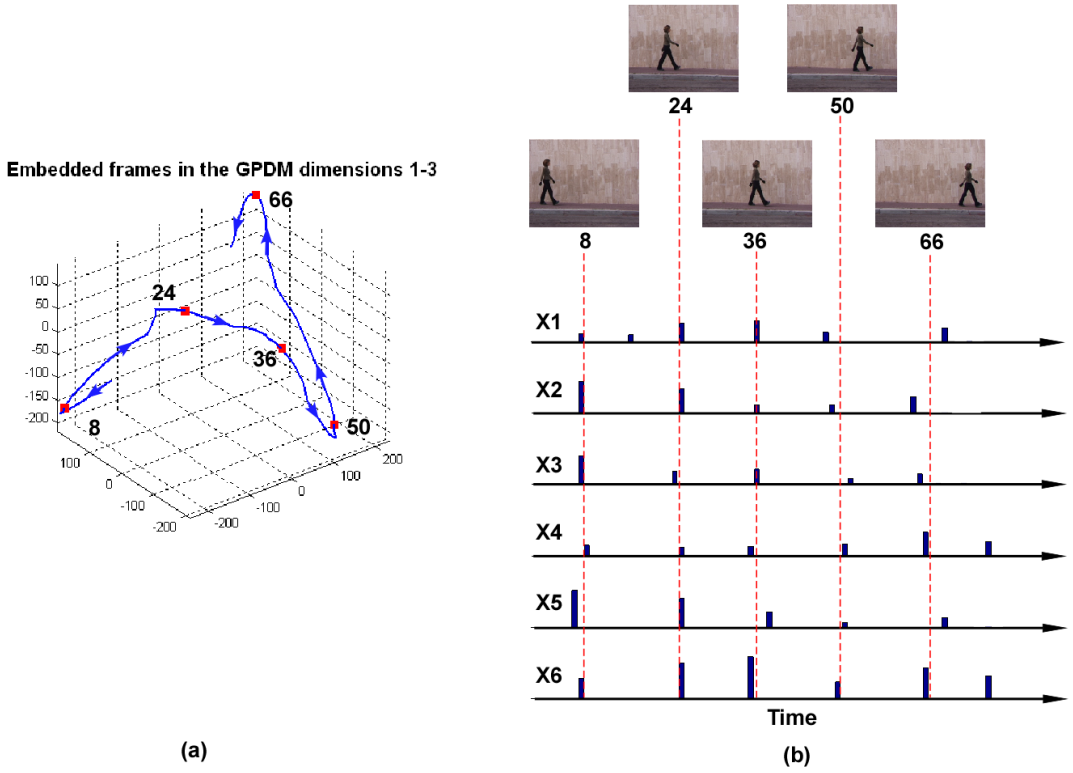


Figure 6.2: (a) The motion trajectory of a video in the Weizmann dataset [BGS+05] in the GPDM dimensions 1-3 (blue curve). (b) The discontinuity in acceleration for variables $x_1, ..., x_6$. Each row represents the locations and the strengths of the local maxima of the 3^{rd} order derivative of a variable. The poses corresponding to the center of the discontinuities (dashed red line) in different variables are shown on top. Frame numbers are displayed both below the poses and in the motion trajectory (red dots).

The Gaussian Process Dynamical Models (GPDM) [WFH08] considers a mapping from a high dimensional data space to a low dimensional latent space and a dynamical model in the latent space. This allows high dimensional data to be compressed in a latent space using Gaussian priors. In [WFH08], the GPDM maps the joint angles in human MoCap data to a 3D latent space.

The relation between the latent variables $\mathbf{X}$ and the original data $\mathbf{Y}$ is as follows:

$$p(\mathbf{X}, \mathbf{Y}, \overline{\alpha}, \overline{\beta}, W) = p(\mathbf{Y}|\mathbf{X}, \overline{\beta}, W)p(\mathbf{X}|\overline{\alpha})p(\overline{\alpha})p(\overline{\beta})p(W) \tag{6.13}$$

where $\overline{\alpha}$, $\overline{\beta}$, and W are the parameters. Given $\mathbf{Y}$, one estimates $\mathbf{X}$, $\overline{\alpha}$, $\overline{\beta}$, and W. Refer to

[WFH08] for details.

We first apply the GPDM to videos. A high dimensional vector of image intensities in each frame is formed by concatenating the pixel values from left to right and then from top to bottom in each image. The GPDM embeds the sequence of intensity vectors in a low dimensional latent space. In our experiment, 6D is generally sufficient. Fig. 6.2 visualizes the trajectory of the latent variables for a video in the GPDM for dimensions 1-3. Then, the alignment algorithm of Sec. 6.3.1 is used for locating the discontinuities.

Our approach does not require foreground/background segmentation. The dimension reduction algorithm handles the smooth background change between two subsequent frames implicitly, while still retaining the changes in acceleration of the human actor. In the Sec. 6.4.2, we further suggest that a background subtraction may not be helpful for our algorithm.

6.4 Experiments

Experiments show that our algorithm finds consistent human poses in MoCap data and in videos, and this consistency facilitates action recognition.

First, we test the algorithm on 30 actions from the CMU MoCap dataset [Lab] for which both MoCap data and videos have been recorded. The evaluation shows that the action discontinuities in motoric space and in visual space are consistent. Second, we show that the detection for action discontinuities in videos outperforms a baseline algorithm, and that the approach is robust to the changes in viewpoint and background. Finally, we demonstrate the potential of our method for recognition by showing that the representation based on action discontinuities improves the performance of action recognition algorithm in [LJD09].

Our approach works for a reasonable range of parameter values, because the localization of the position of the local extrema is not sensitive. In all experiments, we resize the image frames to have the same height (50 pixels) while preserving the height/width ratio for dimension reduction. The parameters for filtering the MoCap data and the reduced visual signal depend on their frame rates. The distance measurement, σ in Eq. 6.6, is fixed to 10 in all experiments.

Our algorithm is also very efficient. L-BFGS is a very fast algorithm. The alignment

algorithm only takes approximately a second on average for a one-minute video in a laptop with 2GHz CPU and 2GB memory, and the current GDPM implementation [WFH08] takes one minute on average.

6.4.1 Effective Detection of Action Discontinuities in Video

In this section, we demonstrate the effectiveness of the action discontinuities detection in videos.

6.4.1.1 Results on the CMU Dataset

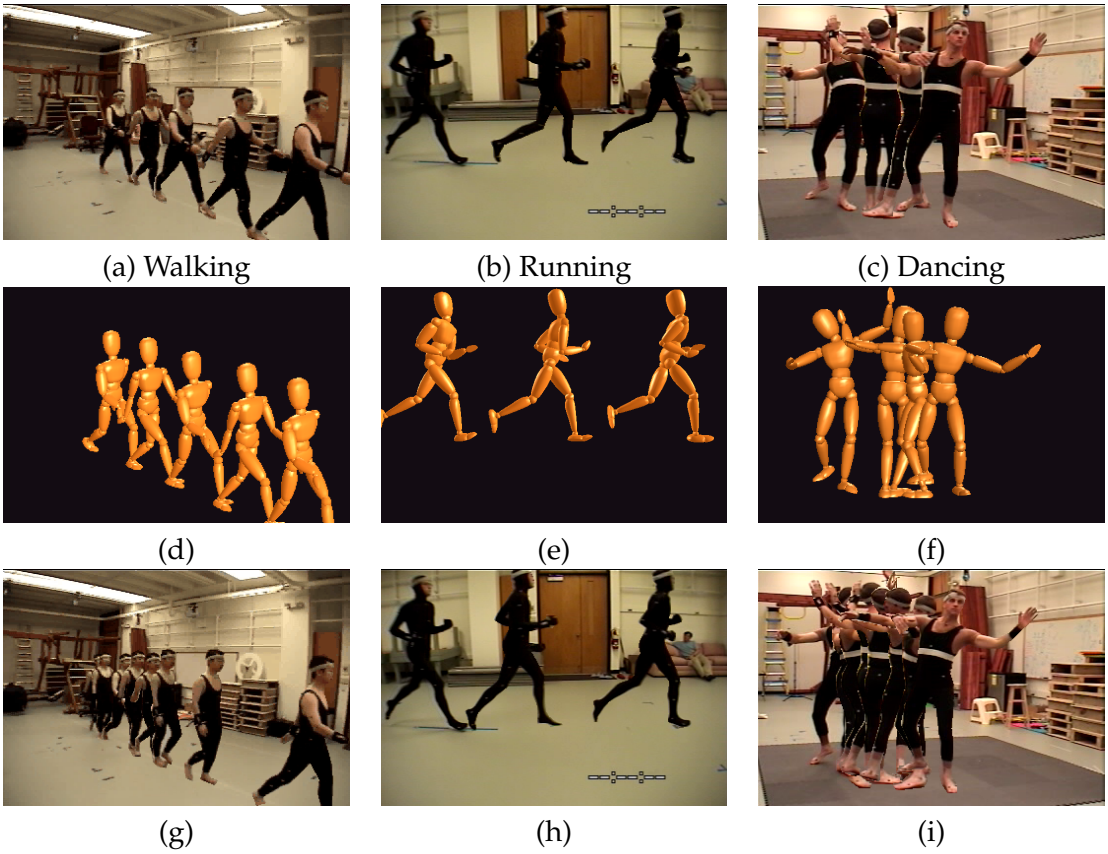
We evaluated the performance of our algorithm on the CMU dataset using simultaneous MoCap data and videos. Our goal is to directly compare whether changes in joint accelerations do correspond to jerk extrema in the signals in the reduced visual space learned from video.

We first selected 30 sequences that have both MoCap data and video available. The actions are "running" (10), "walking" (10), and sport activities (10, including "boxing", "swordplaying", "jumping", etc.). Then we ran the algorithms on the MoCap data and the videos, respectively. Fig. 6.3a-6.3c show the results of the action discontinuities detection in videos for the activities "walking", "running", and "dancing", and Fig. 6.3d-6.3f show the results for the corresponding MoCap data. We overlaid the human subjects for the purpose of visualization.

From Fig. 6.3 we can see that the poses for the videos correspond to the ones in simultaneous MoCap data. This suggests that the dimension reduction is capable of encoding the dynamics of human activities in the reduced space.

The videos and the MoCap data in the CMU dataset were not synchronized. Therefore, in order to evaluate the consistency, we downsampled the MoCap data to 25 fps, and then aligned the results for videos and for MoCap data using dynamic programming. The consistency is defined as the zero-mean standard deviation of the differences between corresponding discontinuities.

The first row of Table 6.1 shows the consistency for the three categories of actions. On average, the standard deviation is within 4 frames (≈ 0.16 second). This consistent result shows that the discontinuities in videos are closely related to the changes in the underlying human body dynamics.



(a) Walking (b) Running (c) Dancing

(d) (e) (f)

(g) (h) (i)

Figure 6.3: a)-c) Results for action discontinuities in videos; d)-f) Results for action discontinuities in corresponding MoCap data; g)-i) Results of the baseline algorithm. Comparing the figures in each column, we clearly see that the action discontinuities for videos correspond to the ones in MoCap data, and that the results for the baseline algorithm are not consistent.

Table 6.1: Consistency between action discontinuities in videos and MoCap data. The consistency is defined as the zero-mean standard deviation of the differences between corresponding discontinuities after applying dynamic programming (fps=25).

Algorithm	Walking	Running	Sport Activities
Proposed Algorithm	2.3	3.1	4
Baseline	10.1	6.4	9.5

6.4.1.2 Evaluation

First, we compared the performance of our algorithm on the selected sequences in the CMU dataset to a baseline algorithm. Then, we evaluated different dimension reduction algorithms using labeled sequences in the UMD Gesture dataset [LJD09].

We used a baseline algorithm [OKA06] for comparison. This algorithm uses the local extrema of the change in the optical flow to compute the action discontinuities. The results show that the baseline algorithm is useful for the actions whose trajectories are parallel to the image plane (Fig. 6.3b) where the perspective projection can be approximated by orthographic projection. However, it may not provide consistent results for the cases where the perspective distortion is severe (Fig. 6.3a) or when the actions are complicated (Fig. 6.3c).

We used the results for MoCap data as the ground truth. The second row in Table. 6.1 further shows that the standard deviations of the baseline algorithm are much larger. The results show that our algorithm is effective and outperform the baseline algorithm.

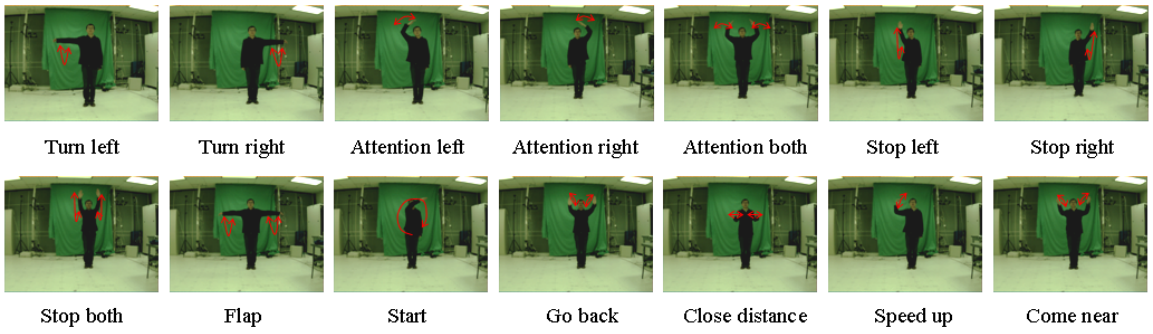


Figure 6.4: UMD Gesture dataset [LJD09]. The dataset contains videos of 14 different gestures of military signals.

Finally, we evaluated the accuracy of our detection algorithm on manually labeled data. The UMD Gesture dataset consists of 14 different gestures of military signals (Fig. 6.4). Each gesture was performed by three subjects, and the same gesture was repeated three

times by each subject. In the dataset, the first and the last pose of each action were manually labeled. This allows us to quantitatively evaluate the accuracy in the discontinuities detection.

The estimated discontinuities naturally divide the video sequences into intervals. We computed the optimal one-to-one-mapping between the estimated intervals and the ground truth intervals. Then the accuracy is defined as the sum of the overlap between corresponding pairs divided by the total length of the intervals. This "assignment" problem can be solved by the Hungarian algorithm (see [YZDJ08] for details). Three other algorithms, namely, PCA, Laplacian [Niy03], and Isomap [TdSL00] are evaluated for comparison.

Compared to the ground truth, Table 6.2 shows that the GPDM has a better accuracy than other algorithms. This result suggests that the GPDM models the process of human activities in videos better than other dimension reduction algorithms which are based on the pairwise relationship of data samples. Therefore, it is more suitable for human motion analysis in a reduced dimension.

This set of experiments demonstrates that action discontinuity detection in video is closely related to the physical dynamics and visual perception of human actions. As a result, the poses at the computed discontinuities are useful representations for processing human actions.

Table 6.2: Accuracies of the detection algorithms using different dimension reduction methods. The accuracy is computed using the Hungarian algorithm on the UMD Gesture dataset [LJD09].

Algo	Ours	PCA	Laplacian	ISOMAP
Rate	**92.3%**	78.6%	85.7%	87.9%

6.4.2 Robust Detection of Action Discontinuities in Videos

In this section, we demonstrate that our proposed algorithm produces robust results under changes in viewpoint and smooth changes in background. Then in Sec. 6.4.3, we will show how to use these results for action recognition.

6.4.2.1 Robustness to Changes in Viewpoint

We show that our detection algorithm provides consistent results on actions viewed from different directions. This makes the results useful for representing actions in multiple view

videos.

We tested our algorithm on synchronized cameras. Fig. 6.5 shows an action from the UMD Pose dataset [OKA06]. The walking sequences of the same person were taken by six synchronized cameras in an indoor environment.

Figure 6.5: Results for a multiple view sequence from the UMD Pose dataset [OKA06]. Each column displays the poses from a different viewpoint. Each row suggests that the estimated key poses from different cameras are consistent. The camera (C#) and the frame number (F#) are shown in the lower right corner in each representative frame.

Fig. 6.5 shows the results for all the six cameras. Each column displays the estimated poses from a different viewpoint. Comparing the frame numbers in individual rows, one can see that the action poses from different cameras are consistent. The reason for this consistency is probably because all the action discontinuities in different views correspond to the changes in the same human motoric dynamics. Therefore, this consistency and robustness of the algorithm make the detection results useful for action recognition.

6.4.2.2 Robustness to Changes in Background

We demonstrate that our approach is robust to smooth background change using videos in public domain. Fig. 6.6a shows an indoor video, which was captured with a moving camera in a gymnastic room. Discontinues in action are correctly identified even though the changes in viewpoints and the background are large. Fig. 6.6b, taken with a moving camera, shows a more challenging situation. The algorithm captures the action discontinuities when the girl moves from left to right and vice versa.

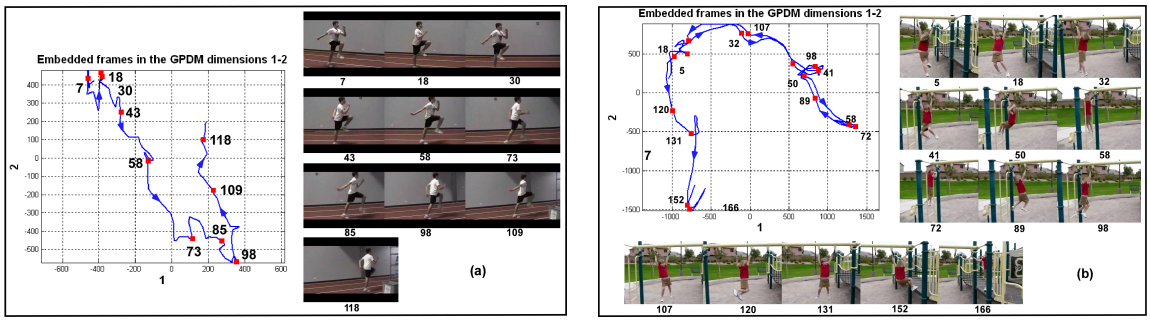


Figure 6.6: Action discontinuities of two videos in public domain. For each action, the motion trajectory (blue curve) of the video in the reduced space is shown. The frame numbers corresponding to the key poses are displayed under the individual frames as well as in the GPDM space 1-2 (red dots). The blue arrow denotes the time direction.

We further demonstrate that background subtraction, even on static cameras, may not be necessary in detecting the action discontinuities. For static cameras, background subtraction can be applied for computing the human masks. In this example, we use the pre-computed mask in the Weizmann dataset to obtain and align the foreground images using correlation, and apply the proposed algorithm on the aligned foreground images. Compared to the poses in Fig. 6.2, the result (Fig. 6.7a) for the masks is not consistent. The performance of discontinuity detection degrades because of inaccurate segmentation and the accumulation of quantization error at the pixel level. This can be easily improved by using optical flow as the input (Fig. 6.7b), which provides similar results as in Fig. 6.2.

Performing background segmentation on videos captured by a moving camera is even more challenging. The current state of the art is to use human body detectors to locate humans and align the centers of the gravity for tracking.

We use an outdoor sequence where the subject is performing the "limping" action. Fig. 6.7c show the results of our algorithm. One can see that the asymmetric walking style is

correctly identified. Fig. 6.7d is the result using a state-of-the-art detector [SKHD09b] as the input. The human detection (yellow bounding boxes) is useful for tracking, but it may not be useful for pose analysis and the detection of the poses that correspond to the action discontinuities.

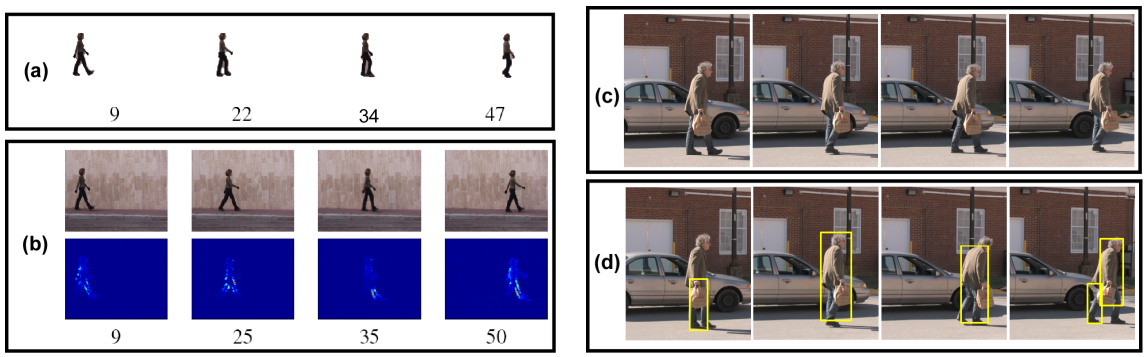


Figure 6.7: (a) and (b): Results for the video in Fig. 6.2 using foreground (a) and flow fields (b) as the input. Frame numbers are displayed below the images. The magnitude of the flow field is shown in (b). (c) Results for a video ("limping" action, taken with moving camera) using our algorithm. This result shows that each step is clearly identified. (d) Action discontinuities based on the human detection results.

The above results show that our algorithm for videos is robust against changes in background. Such consistent results can be adopted for pose and action recognition. In addition, they may be used for action synopsis and representation.

6.4.3 Action Discontinuities Facilitate Visual Action Recognition

In this experiment, we show that the poses corresponding to the action discontinuities lead to accurate action recognition. The primary reason is that these "key poses" reduces the uncharacteristic images in both the training and test datasets. By removing the uncharacteristic poses that are not unique to individual categories, the inter-class variation is increased.

We evaluated the action recognition performance by comparing the results against the algorithm in [LJD09], which uses different sets of poses as input. In our experiments, we employ shape, color, and motion cues of the key poses. Three datasets – the UMD Gesture dataset, the KTH action dataset, and the Weizmann datasets – were used for comparison.

Results show that the poses extracted from our algorithm improve the performance of action recognition (Table. 6.3). For each dataset, we compared the recognition rates using the key poses only and using all cues together, respectively. For the UMD Gesture dataset,

we process them in a slightly different way than in the original paper – we use all the three action instances in each video as the input for the alignment algorithm, instead of using each instance separately.

Table 6.3: Improving the recognition performance using action discontinuities. The results for [LJD09] are copied from the original paper.

Dataset	[LJD09], Pose only	Ours, Pose only	[LJD09], All Cues	Ours, All Cues
UMD	53.57%	**69.44%**	91.07%	**94.44%**
Weizmann	81.11%	**83.33%**	100%	**100%**
KTH S1	71.95%	**76.46%**	98.83%	**99.04%**
KTH S2	61.33%	**69.74%**	94%	**94.52%**
KTH S3	53.03%	**60.13%**	94.78%	**96.32%**
KTH S4	57.36%	**65.54%**	95.48%	**96%**

In Table. 6.3, the recognition rates using the key poses are consistently higher than the other method. This demonstrates that the key poses are useful for action recognition, which in turn suggests that our modeling of action discontinuities in videos is practical.

6.5 Chapter Conclusion

We proposed an action-independent algorithm for detecting human movement discontinuities (defined as the maxima of the third order derivatives of time series) in video and in MoCap data based on the dynamics of human motion. Experiments demonstrate that our algorithm is useful for extracting consistent poses from different videos, and improves the performance of recognition algorithms by providing more discriminative poses as inputs.

The current approach is designed for a single subject in an uncluttered scene. Therefore, it is limited in its abilities at handling complicated scenes of human interactions or activities of a group of subjects. We may be able to handle such scenes using better human visual filters that label the pixels of the body parts. This way, a more refined part based action analysis may be used.

Chapter 7

Conclusion

As seeing agents, human beings have the basic ability to collaborate with other human beings in daily tasks such as cooking, crafting, and cleaning. Action mirroring is a low-level mechanism that allows agents to reproduce 3D human poses directly from 2D videos. In this dissertation, I performed a number of studies to advance the understanding of the computational methods of this mechanism.

First, I suggest that the mapping between the 3D trajectories of a body joint and their 2D projections in video can be learned using a statistical method called Partial Least Squares (PLS). This mirroring module can infer the 3D positions of the body joint from visual data instantly. Synthetic and real experiments showed that the regression framework can robustly mirror human actions.

Second, I studied the primitives in 3D MoCap data of individual human subjects. In our learning procedure, we solve for both the basis functions of the actions and the times when these functions are "activated". The sparse activations explicitly express the coordination among different joints.

Third, I used the Granger Causality to analyze the MoCap data of a group of human subjects in an orchestra. Evidence shows a causality network among human motions in a coordinated setting.

Prof. Rosenfeld once said that we need to "See Far Away"[1]. Therefore, I would like to conclude this dissertation by showing rising areas and applications for computer vision and human action analysis.

7.1 What is Next?

I have a dream that one day Cognitive Robots will rise up, and be our friends. There are two applications directly related to this dream: robots and blind patients.

[1]This is possibly why he named his center "CfAR".

7.1.1 Robots Need Vision

We want to build intelligent machines. Among all components, human action analysis is critical because it facilitates human-robot interaction and robot-object interaction. As Aristotle said, humans are social animals. We cannot survive without interaction, so do cognitive robots.

My PhD study in Maryland will certainly help towards building these humanoid robots. Imitating human action and designing immediate control strategy for response will both entertain users in living room and help them working in the garden. Certainly, this long term goal of Cognitive Robots is still elusive, but let us believe we will see prototypes in the near future.

7.1.2 Blind Patients Need Vision

"There will be a light!"

This is what we are telling the blinds nowadays, and it will eventually come true in 10 years. Bionic eye becomes reality recently, and human trials were carried out in UK, US, and Australia.

However, this visual restoration is limited to a very low resolution. The visual representation is, and will still be, approximately 32 by 32 pixels. Therefore, we must use computer vision algorithms to extract useful information from the environment. As a result, human actions analysis, scene and context understanding, and saliency detection will be very critical. Visual Processing for Bionic Eye, which is the project I am currently working on , will be a useful attempt to see how we merge human with machines.

Appendix A

Publications

1. *Y. Li*, C. Teo, C. Fermuller, and Y. Aloimonos, "Learning Visual-Motoric Mappings of Actions", submitted to *Computer Vision and Pattern Recognition (**CVPR**)*, 2011.

2. *Y. Li*, C. Teo, Y. Yang, C. Fermuller, and Y. Aloimonos, "Action Mirroring for Home Assistant Robots", submitted to *Intl. Conf. on Robotics and Automation (**ICRA**)* 2011.

3. C. Teo, *Y. Li*, Y. Yang, C. Fermuller, and Y. Aloimonos, "Where is My Tool?", submitted to *Intl. Conf. on Robotics and Automation (**ICRA**)* 2011.

4. G. Zhu, X. Yu, *Y. Li*, and D. Doermann, "Language Identification for Handwritten Document Images Using A Shape Codebook", *Pattern Recognition (**PR**)*, 2008.

5. *Y. Li*, Y. Zheng, D. Doerman, and S. Jaeger, "Script-Independent Text Line Segmentation in Freestyle Handwritten Documents", *IEEE Trans. Pattern Anal. Machine Intell. (**T-PAMI**)*, vol. 30, no. 8, pp. 1313-1329, August, 2008.

6. *Y. Li*, Z. Wang, and H. Zeng, "Correlation Filter: An Accurate Approach to Detect and Locate Low Contrast Character Strings In Complex Table Environment:, *IEEE Trans. Pattern Anal. Machine Intell. (**T-PAMI**)*, vol. 26, no. 12, pp. 1639-1644, December, 2004.

7. *Y. Li*, C. Fermuller, and Y. Aloimonos, "Learning Shift-Invariant Sparse Representation of Actions", *Computer Vision and Pattern Recognition (**CVPR**) 2010*.

8. *Y. Li*, C. Fermuller and Y. Aloimonos, "Illusory lightness perception due to signal compression and reconstruction", *Vision Sciences Society (**VSS**) Conference 2010*.

9. *Y. Li*, K. Bitsakos, C. Fermuller, and Y. Aloimonos, "Real-Time Shape Retrieval for Robotics Using Skip Tri-Grams", In Proc. of *Intl. Conf. on Intell. Robots and Sys. (**IROS**) 2009*.

10. G. Zhu, X. Yu, *Y. Li* and D. Doermann, "Learning Visual Shape Lexicon for Document Image Content Recognition", In Proc. of *European Conf. on Computer Vision (**ECCV**) 2008*, pp. 745-758.

11. K. Bitsakos, H. Yi, *Y. Li*, and C. Fermuller, "Bilateral symmetry of object silhouettes under perspective projection", In Proc. of *Intl. Conf. on Pattern Recognition (**ICPR**) 2008*.

12. *Y. Li* and D. Jacobs, "Efficiently Determining Silhouette Consistency", In Proc. of *Computer Vision and Pattern Recognition (**CVPR**) 2007*.

13. X. Yu, *Y. Li*, C. Fermuller, and D. Doermann, "Object Detection Using A Shape Codebook", In Proc. of *British Machine Vision Conference (**BMVC**) 2007*.

14. K. Bitsakos, *Y. Li*, and C. Fermuller, "Combining Motion from Texture and Lines for Visual Navigation", In Proc. of *Intl. Conf. on Intell. Robots and Sys. (**IROS**) 2007*, pp. 232-239.

15. *Y. Li*, Y. Zheng and D. Doermann, "Detecting Text Line in Handwritten Documents". In Proc. of *Intl. Conf. on Pattern Recognition (**ICPR**) 2006*, pp 1030-1033.

16. *Y. Li*, Z. Wang, and H. Zeng, "String Extraction From Color Airline Coupon Image Using Statistical Approach", In Proc. of *Intl. Conf. on Doc. Anal. and Rec. (**ICDAR**) 2003*, pp. 289-294.

Appendix B

Brief Resume

B.1 Education

- 2004-present, **PH.D** student in Electrical and Computer Engineering
  University of Maryland, College Park, MD 20742.
- 2001-2004, **M.ENG** in Computer Science (First Rank),
  South China Univ. of Tech., China.
- 1998-2001, **B.SC** in Computer Science (First Honor),
  South China Univ. of Tech., China.

B.2 Awards and Honors

- 2008-present, **FUTURE FACULTY FELLOW**,
  A. James Clark School of Engineering, **University of Maryland**.
- 2007, **SECOND PLACE**, the **First Semantic Robot Vision Challenge**,
  AAAI 2007, Vancouver, Canada.
- 2006**BEST STUDENT PAPER**,
  the *Intl. Conf. on Frontier in Handwriting Recognition (**ICFHR**)* 2006.

B.3 Research Experience

- 2007-present, **RESEARCH ASSISTANT**,
  Computer Vision Lab, **University of Maryland**.
- 2004-2007, **RESEARCH ASSISTANT**,
  Language and Media Lab, **University of Maryland**.
- 2001-2004, **STUDENT TECHNICAL MEMBER**,
  Huagong Tomorrow Co. Ltd., China.

B.4 Participated Projects

- 2010-present, **VISUAL PROCESSING FOR BIONIC EYE**,
  National ICT Australia and Bionic Vision Australia.
- 2007-present, **POETICON**,
  European Commission FP7.
- 2007-2008, **PARKINSON'S DISEASE ANALYSIS**,
  National Institutes of Health.
- 2004-2007, **CONTRACT MDA-9040-2C-0406**,
  Department of Defense.

Bibliography

[Abd07] H. Abdi. *Partial Least Squares (PLS) Regression.* Thousand Oaks, 2007.

[ACCO05] Jackie Assa, Yaron Caspi, and Daniel Cohen-Or. Action synopsis: pose selection and illustration. *ACM Trans. Graph.*, 24(3):667–676, 2005.

[AGC10] Giorgio Metta Arjan Gijsberts, Tatiana Tommasi and Barbara Caputo. Object recognition using visuo-affordance maps. In *IROS*, 2010.

[Ari] Aristotle. *On the Gait of Animals*.

[Ari06] Okan Arikan. Compression of motion capture databases. *ACM Trans. Graph.*, 25(3):890–897, 2006.

[ARS09] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *CVPR*, pages 1014–1021, 2009.

[AT06] Ankur Agarwal and Bill Triggs. Recovering 3d human pose from monocular images. 28(1):44–58, 2006.

[BD06] Thomas Blumensath and Mike E. Davies. Sparse and shift-invariant representations of music. *IEEE Transactions on Audio, Speech & Language Processing*, 14(1):50–57, 2006.

[BGS+05] Moshe Blank, Lena Gorelick, Eli Shechtman, Michal Irani, and Ronen Basri. Actions as space-time shapes. In *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision*, pages 1395–1402, Washington, DC, USA, 2005. IEEE Computer Society.

[Bis05] A. Bissacco. Modeling and learning contact dynamics in human motion. In *CVPR'05*, pages 421–428, 2005.

[BJ90] George Edward Pelham Box and Gwilym Jenkins. *Time Series Analysis, Forecasting and Control*. Holden-Day, Incorporated, 1990.

[BJ01] Y.Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary region segmentation of objects in n-d images. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 105–112 vol.1, 2001.

[BM10] T. Brox and J. Malik. Large displacement optical flow: Descriptor matching in variational motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 99(PrePrints), 2010.

[Bou08] D. Bouchard. *Automated Motion Capture Segmentation using Laban Movement Analysis*. PhD thesis, University of Pennsylvania, 2008.

[Bra99] R. Bracewell. *The Fourier Transform & Its Applications*. McGraw-Hill Science, June 1999.

[Can06] Emmanuel Candes. Compressive sampling. *Int. Congress of Mathematics*, pages 1433–1452, 2006.

[CDS01] Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders. Atomic decomposition by basis pursuit. *SIAM Rev.*, 43(1):129–159, 2001.

[CGM+] R. Chalodhorn, D. Grimes, G. Maganis, R. Rao, and M. Asada. Learning humanoid motion dynamics through sensory-motor mapping in reduced dimensional spaces. In *ICRA'06*.

[CJ06] M. Chhabra and R. Jacobs. Properties of synergies arising from a theory of optimal motor behavior. *Neural Comput.*, 18(10):2320–2342, 2006.

[CJLS09] J.F. Cai, H. Ji, C.Q. Liu, and Z.W. Shen. Blind motion deblurring from a single image using sparse approximation. pages 104–111, 2009.

[Cle94] W. S. Cleveland. Coplots, Nonparametric Regression, and Conditionally Parametric Fits. In T. W. Anderson, K. T. Fang, and I. Olkin, editors, *Multivariate Analysis and Its Applications*, pages 21–36. IMS Lecture Notes — Monograph Series, Vol. 24, Hayward, California, 1994.

[CVV09] Antonio Camurri, Giovanna Varni, and Gualtiero Volpe. Measuring entrainment in small groups of musicians. In *ACII '09*, 2009.

[ddSB03] A. d'Avella d'Avella, P. Saltiel, and E. Bizzi. Combinations of muscle synergies in the construction of a natural motor behavior. *Nature Neuroscience*, 6:300–308, 2003.

[Deb78] C. Deboor. *A Practical Guide to Splines*. Springer-Verlag Berlin and Heidelberg GmbH & Co. K, December 1978.

[DGJL07] Alexandre D'aspremont, Laurent E. Ghaoui, Michael I. Jordan, and Gert R. G. Lanckriet. A direct formulation for sparse pca using semidefinite programming. *SIAM Review*, 49(3):434–448, 2007.

[DGL09] Z. Deng, Q. Gu, and Q. Li. Perceptually consistent example-based human motion retrieval. In *I3D '09*, pages 191–198, 2009.

[DHS00] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*, pages 84–97. Wiley-Interscience Publication, second edition, 2000.

[DS98] N. Draper and H. Smith. *Applied Regression Analysis (Wiley Series in Probability and Statistics)*. Wiley-Interscience, third edition, April 1998.

[DT05a] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition (CVPR) 2005*, pages 886–893, 2005.

[DT05b] David L. Donoho and Jared Tanner. Sparse nonnegative solution of underdetermined linear equations by linear programming. *Proceedings of the National Academy of Sciences*, 102(27):9446–9451, 2005.

[EBMM03] A. A. Efros, A. C. Berg, G. Mori, and J. Malik. Recognizing action at a distance. pages 726–733 vol.2, 2003.

[EL04] Ahmed M. Elgammal and Chan-Su Lee. Inferring 3d body pose from silhouettes using activity manifold learning. In *CVPR (2)*, pages 681–688, 2004.

[FCO05] L. Fadiga, L. Craighero, and E. Olivier. *Current Biology*, 2005.

[FH05] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1):55–79, January 2005.

[FMJZ] V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Progressive search space reduction for human pose estimation. In *CVPR 2008*.

[Fri08] C. Frith. Social cognition. *Philos Trans R Soc Lond B Biol Sci*, pages 2033–2039, 2008.

[GKS02] Shaogang Gong, Jeffrey Ng Sing Kwong, and Jamie Sherrah. On the semantics of visual behaviour, structured events and trajectories of human action. *IVC*, 20(12):873–888, 2002.

[GLL+04] Enrico Giunchiglia, Joohyung Lee, Vladimir Lifschitz, Norman McCain, and Hudson Turner. Nonmonotonic causal theories. *Artif. Intell.*, 153(1-2):49–104, 2004.

[GO09] Tom Goldstein and Stanley Osher. The split bregman method for l1-regularized problems. *SIAM J. on Imaging Sciences*, 2(2):323–343, 2009.

[Gra69] Clive W. J. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37(3):424–438, July 1969.

[Hay] Monson H. Hayes. *Statistical Digital Signal Processing and Modeling*. Wiley.

[JM02] Odest Chadwicke Jenkins and Maja J Mataric. Deriving action and behavior primitives from human motion data. In *In International Conference on Intelligent Robots and Systems*, pages 2551–2556, 2002.

[Joh73] G. Johansson. Visual perception of biological motion and a model for its analysis. *Perception and Psychophysics*, 14:201–211, 1973.

[Kan87] Immanuel Kant. *Critique of Pure Reason*. 1787.

[KSH] Yan Ke, Rahul Sukthankar, and Martial Hebert. Spatio-temporal shape and flow correlation for action recognition. *CVPR'07*.

[KWC] Tae-Kyun Kim, Shu-Fai Wong, and Roberto Cipolla. Tensor canonical correlation analysis for action classification. *CVPR'07*.

[Lab] CMU Graphics Lab. http://mocap.cs.cmu.edu.

[LDB+09] A. Londei, A. D'Ausilio, D. Basso, C. Sestieri, C. Gratta, G. Romani, and M. Belardinelli. Sensory-motor brain network connectivity for speech comprehension. *Hum Brain Mapp.*, 2009.

[LF04] ChunMei Lu and Nicola J. Ferrier. Repetitive motion analysis: Segmentation and event classification. 26(2):258–263, 2004.

[LFAJ10] Y. Li, C. Fermuller, Y. Aloimonos, and H. Ji. Learning shift-invariant sparse representation of actions. In *CVPR*, pages 2630–2637, 2010.

[LJ07] Haibin Ling and David W. Jacobs. Shape classification using the inner-distance. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(2):286–299, 2007.

[LJD09] Zhe Lin, Zhuolin Jiang, and Larry S. Davis. Recognizing actions by shape-motion prototype trees. *CVPR'09*, 0:1–8, 2009.

[LM] G. Liu and L. McMillan. Segment-based human motion compression. In *SCA '06*, pages 127–135.

[LN07] Fengjun Lv and Ramakant Nevatia. Single view human action recognition using key pose matching and viterbi path searching. *CVPR'07*, 0:1–8, 2007.

[LP09] James LeSage and Kelley Pace. *Introduction to Spatial Econometrics*. CRC Press, 2009.

[lTFHB+] F. De la Torre Frade, J. Hodgins, A. Bargteil, X. Artal, J. Macey, A. Collado I. Castells, and J. Beltran. Guide to the carnegie mellon university multimodal activity (cmu-mmac) database. Technical Report CMU-RI-TR-08-22.

[Lut07] Helmut Lutkepohl. *New Introduction to Multiple Time Series Analysis*. Springer, 1st ed. 2006. corr. 2nd printing edition, July 2007.

[MHS] Pyry Matikainen, Martial Hebert, and Rahul Sukthankar. Representing pairwise spatial and temporal relations for action recognition. In *ECCV 2010*.

[MOV] MOVEN. *http://www.moven.com/*.

[MPK] Ross Messing, Chris Pal, and Henry Kautz. Activity recognition using the velocity histories of tracked keypoints. In *ICCV '09*.

[MRC+07] M. Müller, T. Röder, M. Clausen, B. Eberhardt, B. Krüger, and A. Weber. Documentation mocap database hdm05. Technical Report CG-2007-2, Universität Bonn, June 2007.

[MRPK09] C. Maidhof, M. Rieger, W. Prinz, and S. Koelsch. Nobody is perfect: ERP effects prior to performance errors in musicians indicate fast monitoring processes. *PLoS One*, page e5032, 2009.

[MSN+06] G. Metta, G. Sandini, L. Natale, L. Craighero, and L. Fadiga. Understanding mirror neurons: A bio-robotic approach. In *Interaction Studies*, pages 197–232, 2006.

[MT97] Norman McCain and Hudson Turner. Causal theories of action and change. In *AAAI*, pages 460–465, 1997.

[Muy] Eadweard Muybridge. *The Human Figure in Motion*.

[MV82] D. Marr and L. Vaina. Representation and recognition of the movements of shapes. *Proceedings of the Royal Society of London B*, 214:501–524, 1982.

[Niy03] Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15:1373–1396, 2003.

[NS01] Arnold Neumaier and Tapio Schneider. Estimation of parameters and eigenmodes of multivariate autoregressive models. *ACM Trans. Math. Softw.*, 27(1):27–57, 2001.

[OKA06] Abhijit S. Ogale, Alap Karapurkar, and Yiannis Aloimonos. View-invariant modeling and recognition of human actions using grammars. In *WDV*, pages 115–126, 2006.

[Par62] Emanuel Parzen. On estimation of a probability density function and mode. *Ann. Math. Stat*, (33):1065–1076, 1962.

[Pea00] Judea Pearl. *Causality : Models, Reasoning, and Inference*. Cambridge University Press, 2000.

[PHRA] N. Pollard, J. Hodgins, M. Riley, and C. Atkeson. Adapting human motion for the control of a humanoid robot. In *ICRA '02*.

[Ple03] Robert Pless. Image spaces and video trajectories: Using isomap to explore video sequences. *ICCV'03*, 2:1433, 2003.

[PRAP08] Yael Pritch, Alex Rav-Acha, and Shmuel Peleg. Nonchronological video synopsis and indexing. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(11):1971–1984, 2008.

[PSZ08] S. Paisitkriangkrai, C. Shen, and J. Zhang. Fast pedestrian detection using a cascade of boosted covariance features. *IEEE TCSVT*, 18(8):1140–1151, 8 2008.

[Qua] *http://en.wikipedia.org/wiki/Quaternion*.

[RA] Y. Rui and P. Anandan. Segmenting visual actions based on spatio-temporal motion patterns. In *CVPR'00*, pages I: 111–118.

[Ram06] D. Ramanan. Learning to parse images of articulated bodies. *Advanced in Neural Information Processing Systems*, 2006.

[RC04] G. Rizzolatti and L. Craighero. The mirror-neuron system. *Annu Rev Neurosci*, 27:169–192, 2004.

[RKB04] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. "grabcut": interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3):309–314, 2004.

[RKN07] Roger Grosse Rajat Raina, Helen Kwong, and Andrew Y. Ng. Shift-invariant sparse coding for audio classification. In *UAI*, 2007.

[RS] M. Raptis and S. Soatto. Tracklet descriptors for action modeling and video analysis. In *ECCV 2010*.

[RSH+05] Liu Ren, Gregory Shakhnarovich, Jessica K. Hodgins, Hanspeter Pfister, and Paul A. Viola. Learning silhouette features for control of human motion. *ACM Trans. Graph.*, 24(4):1303–1331, 2005.

[RW05] Carl E. Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, December 2005.

[SB06] L. Sigal and M. J. Black. Humaneva: Synchronized video and motion capture dataset for evaluation of articulated human motion. *Brown Univertsity Technical Report*, 2006.

[SC08] Aravind Sundaresan and Rama Chellappa. Model driven segmentation of articulating humans in laplacian eigenspace. 30(10):1771–1785, 2008.

[Sca02] Brian Scassellati. Theory of mind for a humanoid robot. *Autonomous Robots*, 12(1):13–24, January 2002.

[SE07] Anil K. Seth and Gerald M. Edelman. Distinguishing causal interactions in neural populations. *Neural Computation*, 19(4):910–933, April 2007.

[SHK09] Takashi Shibuya, Tatsuya Harada, and Yasuo Kuniyoshi. Causality quantification and its applications: structuring and modeling of multivariate time series. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 787–796, 2009.

[SKHD09a] W. Schwartz, A. Kembhavi, D. Harwood, and L. Davis. Human detection using partial least squares analysis. In *ICCV*, 2009.

[SKHD09b] W.R. Schwartz, A. Kembhavi, D. Harwood, and L.S. Davis. Human detection using partial least squares analysis. In *Accepted to be presented in the International Conference on Computer Vision*, 2009.

[SLC04] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local svm approach. In *ICPR'04*, volume 3, pages 32–36 Vol.3, 2004.

[SMI07] S. Schaal, P. Mohajerian, and A. Ijspeert. Dynamics systems vs. optimal control a unifying view. *Progress in Brain Research*, 165:425–445, 2007.

[SMV07] G. Sandini, G. Metta, and D. Vernon. The iCub Cognitive Humanoid Robot: An Open-System Research Platform for Enactive Cognition. In *50 Years of Artificial Intelligence*, Lecture Notes in Computer Science, chapter 32, pages 358–369. 2007.

[SNI04] Takaaki Shiratori, Atsushi Nakazawa, and Katsushi Ikeuchi. Detecting dance motion structure through music analysis. *In AFGR'04*, 0:857, 2004.

[Ste08] Keith Stenning. *Human Reasoning and Cognitive Science*. Cambridge University Press, 2008.

[STT10] B. Sapp, A. Toshev, and B. Taskar. Cascaded models for articulated pose estimation. In *ECCV*, 2010.

[SvG] K. Schindler and L. van Gool. Action snippets: How many frames does human action recognition require? In *CVPR'08*.

[TBB] Moritz Tenorth, Jan Bandouch, and Michael Beetz. The TUM Kitchen Data Set of Everyday Manipulation Activities for Motion Tracking and Action Recognition. In *IEEE Int. Workshop on Tracking Humans for the Evaluation of their Motion in Image Sequences (THEMIS). In conjunction with ICCV 2009*.

[TdSL00] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, December 2000.

[TG07] J. A. Tropp and A. C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on Information Theory*, 53(12):4655–4666, 2007.

[TRE] *http://www-nlpir.nist.gov/projects/trecvid*.

[UD08a] R. Urtasun and T. Darrell. Sparse probabilistic regression for activity-independent human pose inference. In *CVPR'08*, pages 1–8, 2008.

[UD08b] Raquel Urtasun and Trevor Darrell. Sparse probabilistic regression for activity-independent human pose inference. In *CVPR*, 2008.

[VIC] VICON. *http://www.vicon.com/*.

[VMP03] D. Vecchio, R. Murray, and P. Perona. Decomposition of human motion into dynamics based primitives with application to drawing tasks. *Automatica*, 39:2085–2098, 2003.

[WFH08] Jack M. Wang, David J. Fleet, and Aaron Hertzmann. Gaussian process dynamical models for human motion. 30(2):283–298, 2008.

[WWW08] Xiaozhe Wang, Liang Wang, and Anthony Wirth. Pattern discovery in motion time series via structure-based spectral clustering. In *CVPR'08*, 2008.

[XG08] Tao Xiang and Shaogang Gong. Video behavior profiling for anomaly detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(5):893–908, 2008.

[YJSB09] Yang, R. Jafari, S. S. Sastry, and R. Bajcsy. Distributed recognition of human actions using wearable motion sensor networks. *Ambient Intelligence and Smart Environments*, 2009.

[YZDJ08] L. Yi, Y. Zheng, D. Doermann, and S. Jaeger. Script-Independent Text Line Segmentation in Freestyle Handwritten Documents. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2008.

[Zat97] V. Zatsiorsky. *Kinematics of Human Motion*. Human Kinetics Publishers, September 1997.