2013

Modeling and Recognizing Binary Human Interactions

Ke Feng
West Virginia University

Modeling and Recognizing Binary Human Interactions

by

Ke Feng

Thesis submitted to the
College of Engineering and Mineral Resources
at West Virginia University
in partial fulfillment of the requirements
for the degree of

Master of Science
in
Computer Science

Donald A. Adjeroh, Ph.D.
Mark V. Culp, Ph.D.
Gianfranco Doretto, Ph.D., Chair

Lane Department of Computer Science and Electrical Engineering

Morgantown, West Virginia
2013

Keywords: Human interaction recognition, Computer vision, Machine learning, Non-linear dynamical system, Pairwise kernels, Kernel methods, Video analysis

Abstract

Modeling and Recognizing Binary Human Interactions

by

Ke Feng
Master of Science in Computer Science

West Virginia University

Gianfranco Doretto, Ph.D., Chair

Recognizing human activities from video is an important step forward towards the long-term goal of performing scene understanding fully automatically. Applications in this domain include, but are not limited to, the automated analysis of video surveillance footage for public and private monitoring, remote patient and elderly home monitoring, video archiving, search and retrieval, human-computer interaction, and robotics. While recent years have seen a concentration of works focusing on modeling and recognizing either group activities, or actions performed by people in isolation, modeling and recognizing binary human-human interactions is a fundamental building block that only recently has started to catalyze the attention of researchers.

This thesis introduces a new modeling framework for binary human-human interactions. The main idea is to describe interactions with spatio-temporal trajectories. Interaction trajectories can then be modeled as the output of dynamical systems, and recognizing interactions entails designing a suitable way for comparing them. This poses several challenges, starting from the type of information that should be captured by the trajectories, which defines the geometry structure of the output space of the systems. In addition, decision functions performing the recognition should account for the fact that the people interacting do not have a predefined ordering. This work addresses those challenges by carefully designing a kernel-based approach that combines non-linear dynamical system modeling with kernel PCA. Experimental results computed on three recently published datasets, clearly show the promise of this approach, where the classification accuracy, and the retrieval precision are comparable or better than the state-of-the-art.

Acknowledgements

I would like to acknowledge people for helping me during my graduate studies. Firstly, I would especially like to thank my committee chair and research advisor, Dr. Gianfranco Doretto, for his generous time and commitment. Throughout my graduate work he encouraged me to develop independent thinking and research skills. He continually stimulated my analytical thinking and greatly assisted me with scientific writings.

Secondly, I am also very grateful for having an exceptional master thesis committee, and wish to thank Dr. Donald A. Adjeroh and Dr. Mark Culp for their support and encouragement. Especially, Dr. Adjeroh gave me lots of help during my graduate years in this department.

My sincere thanks also go to Dr. James Mooney for offering me a one year research support in his research groups, and leading me working on diverse exciting projects.

In addition I would also like to thank my fellow labmates for their stimulating discussions, assistance, generosity, support and their advice. They are Saied Motiian, Sajid Sharlemin, Harika Bharthavarapu, and Farzad Siyahjani.

Finally, I would like to extend many thanks to my friends, my parents and my wife for their constant help and support throughout all my life.

Contents

List of Figures

List of Tables

Notation

We use the following notation and symbols throughout this thesis.

$\Phi(\cdot)$	:	Mapping function
$\mathcal{S}$	:	Input feature space
$\mathcal{H}$	:	Hilbert space
$\{\cdot\}$	:	Temporal sequence
$E[\cdot]$	:	Expectation operator
$\mathbb{H}$	:	Histogram space
$\mathbb{R}^n$	:	Real space with n dimension
v_t	:	System noise
w_t	:	Observation noise
λ	:	Weight
$\|\cdot\|$	:	Matrix norm
δ	:	Threshold
$\mathbf{y}_{i,j}$	:	Interaction trajectory of i-th person and j-th person
K	:	Kernel
$\mathbf{h}$	:	Histogram of oriented optical flow feature
$\mathbf{m}$	:	Motion Histogram
$(\cdot)^\top$	:	Transpose
$\doteq$	:	Approximately equal
b and τ	:	number of bins

Chapter 1

Introduction

Recognizing human interactions from video is an important step forward towards the long-term goal of performing scene understanding fully automatically. It is applicable to various domains including video surveillance, video annotation, autonomous robotics, video analysis, egocentric activity recognition, etc. The goal of such recognition is to automatically analyze ongoing activities from unknown videos and correctly classify them into activity categories.Human activities can be categorized into four different levels: single person activities, human-object interactions, human-human interactions, and group activities. Single person activities such as “walking”, “kicking”, “dancing” are atomic activities. Binary interactions are human activities that involve two persons such as “shaking hands”, “kissing”, and “hugging”. Group activities are the activities performed by groups composed of multiple persons such as “group walking”, “group waiting”, “queuing”. In recent years, extensive human action recognition works concentrated on the problem of recognizing single-person gestures and actions. Such recognition technology has been applied in many industry areas such as security, surveillance, games, robotics, etc. Besides the recognition of single person activities, group activities recognition also received a lot of attention from researchers. Some promising results are shown in [1, 2, 3]. Compared with the recognition of single person activities and group activities, the area of modeling the interactions between two people is much less explored. Only in the last few years, more realistic interaction datasets [4, 5, 6] have become available. This triggered the development of more sophisticated approaches [4, 7, 5, 8, 6, 9].

In human activity recognition, the study of single person activities reveals each person’s motion and activities in the scene, while the study of binary person interactions indicates the relationship

between two humans in the scene. With the interaction information of each pair of humans, more complicate activities and events could be recognized. Therefore, the study of binary interaction recognition will greatly contribute to the development of artificial intelligence (AI). In order to quickly and accurately recognize binary interactions, it is necessary to establish an efficient modeling framework. This thesis aims at developing such framework leading to an approach that is fast, and that could become a building block for analyzing the behavior of a larger crowd in a scene, monitored by a network of cameras. We assume that people in the scene are been tracked, and the tracking information is known (see Chapter 5 for more detail). This allows to analyze the spatio-temporal volume around each person and to extract relevant motion features. At the same time, the tracking information of a pair of individuals enables the extraction of a set or proximity cues, which coupled with the motion cues form interaction trajectories.

To make such interaction trajectories useful in our framework, this thesis models them as the output of non-linear dynamical systems (NLDS), and therefore reduces the problem of recognizing human interactions to the problem of discriminating between NLDSs. However, this method requires designing special kernels that satisfy certain properties. To do so, this special kernel design has to take into account the geometry of the space where the interaction trajectories live. In addition, some certain symmetry properties, which are induced by the fact that we are modeling people interactions, have to be considered. In this thesis, we addressed both problems by carefully exploiting kernel construction techniques, and by clearly showing that kernels for recognizing interaction trajectories should belong to a subcategory of the so-called pairwise kernels, and in particular they should satisfy the balanced property.

Besides the above mentioned contributions, other contributions of this thesis includes: A description of how human interactions can be represented by interaction trajectories, where we introduce a new efficient motion feature, called motion histogram; A formal setup of the human interaction recognition problem, and the identification of the challenges it implies; A description of how interaction trajectories are represented by NLDSs; The explanation of how to design kernels for comparing interaction trajectories, while addressing the challenges outlined. It is also worth mentioning that a positive side effect of this framework is that by using pairwise symmetric and balanced kernels not only one can boost performance, but also is possible to significantly reduce the training time, since there is no need to use a symmetric training dataset, which has double the

size of a regular one.

This thesis is organized as follows. Chapter 2 gives an overview of human activity recognition and binary interaction recognition. Some basic tools for human action recognition and approaches are also discussed in this chapter. In Chapter 3, a modeling framework of binary interactions is developed and the principles of this framework and application domain are explained in detail. Chapter 4 focuses on pairwise kernel design, whereas Chapter 5 describes the dataset and experimental results. This chapter shows classification, and retrieval experiments where several discussed kernels are tested, validating the proposed framework from the theoretical perspective, as well as practical by achieving very promising results. A comparison between our method and other state-of-art approaches is also performed. In Chapter 6, we give a summary of this thesis and propose some possible future directions of investigation.

Chapter 2

Overiew of Human Activity Recognition

Human action and activity recognition is of significant interest in applications that range from computer game development to public security monitoring. With more and more applications in the computer intelligence area, it has become increasingly important in recent years. This technology of human action and activity recognition was developed and inspired by object recognition techniques. In 1973, Johansson attached lights to major joints of a person in his experiment and analyzed the structure and motion [10]. This probably is the earliest experiment related to human action recognition. In 1982, inspired by Johansson's experiment, Jon Webb and J. K. Aggarwal separate such a motion into a rotation and a translation, where they assume the rotation axis is fixed for short periods of time. So the structure of jointed objects can be determined under orthographic projection [11]. Their works may be considered as the beginning of human action and activity recognition. After the 1980s, this field receives more attention from researchers. Especially in this decade, numerous publications focus on this area.

From different perspectives, human action recognition can be categorized with different taxonomies . If the levels of video understanding is taken into account, it can be separated into four levels [12]: Object-level, Tracking-level, Pose-level, and Activity-level. From video complexity, action recognition can be divided into single person action recognition, human to human interaction (also called as binary interaction) recognition, and group activity recognition, as described in the previous chapter. If considered from the algorithms approach, human action recognition can be categorized as single-layer approaches and hierarchical approaches. This chapter gives a brief description of each classification from these different perspectives as well as the general tools used

Reference	Background subtraction	Human feature
Wren et al. [1997]	Color/Ref. image	Color,contour
Beleznai et al. [2004]	Color/Ref. image	Region model
Haga et al. [2004]	Color/Ref. image	F1-F2-F3
Eng et al. [2004]	Color/Ref. image	Color
Elzein et al. [2003]	Motion/Frame diff.	Wavelets
Toth and Aach [2003]	Motion/Frame diff.	Fourier shape
Lee et al. [2004]	Motion/Frame diff.	Shape
Zhou and Hoang [2005]	Motion/Frame diff.	Shape
Yoon and Kim [2004]	Motion + Color	Geom Pix. Val.
Xu and Fujimura [2003]	Depth	Motion
Li et al. [2004]	Depth	Shape
Han and Bhanu [2003]	Infrared	IR+color
Jiang et al. [2004]	Infrared	IR+color

Table 2.1: Methods using background subtraction [13]

for these recognitions.

2.1 Video-undestanding-based taxonomy of human activity recognition

As mentioned before, human action recognition can be explored from four different levels: Object-level, Tracking-level, Pose-level, and Activity-level. The main issue for the object level is to detect whether a human present at a certain. Typical dataset for this kind of category is pedestrian detection. All people in the given video should be recognized and automatically marked. The algorithms for such detection is the same as the detection of other kinds of objects. These algorithms were classified as “based on background subtraction” and “based on direct detection” [13]. Background subtraction techniques usually have a background reference which can be subtracted from video frames to obtain foreground objects. These objects will be classified as human or other objects based on shape, color, or motion or other features. Direct techniques classify video patches as human or non-human based on both 2D and 3D features. 3D features are extracted from the motion. Table 2.1 and Table 2.2 show the usage of these two methods by some publications, respe‘ctively.

Reference	Human model	Classier
Cutler and Davis [2000]	Periodic Motion	Motion similarity
Utsumi and Tetsutani [2002]	Geom. Pix. Val	Distance
Gavrila and Giebel [2002]	Shape template	Chamfer dist.
Viola et al. [2003]	shape+motion	Adaboost cascade
Sidenbladh [2004	Optical ow	SVM (RBF)
Dalal and Triggs [2005]	Hist. of gradients	SVM (Linear)

Table 2.2: Methods based on direct detection [13]

Tracking, which usually is combined with detection, is another important part in human action recognition. Trajectories can be determined though tracking. Therefore, we are able to obtain the cues of human motion and relationship by analyzing the collection of trajectories in the video.

Besides the trajectories, human pose recognition is also an important aspect for video understanding. Joint location of a person was measured here, and the whole video was considered as a sequence of poses. For certain action categories where trajectory is not sufficient, analysis of human pose provides a better approach for classification. Traditionally, there are two broad classes of approaches for such recognition[14]: One is matching templates which are called as exemplar-based approaches [15, 16, 17, 18]; Another one consists of fitting a human body model[19, 20, 21]. Both approaches were extensively explored in recent years and are successfully applied.

The last level for video understanding is activity level. There are many types of human activities. We can divide these activities to single human actions (include gesture), human human interactions, and group activities. These activities are represented by a collection of human/human body part movements with a particular semantic meaning. A computer will analyze the video recorded by a camera or camcorder and automatically detect the ongoing events from these video data.

2.2 Approach-based taxonomy of human activity recognition

Single layer approaches and hierarchical approach are two methodologies for human activity recognition. In the single layer approaches, human activities are directly recognized based on video data or sequences of images. To do so, low level features are directly extracted from video data. These features are then processed by machine learning technique such as linear support vector

machine (SVM) or hidden Markov model (HMM) to determine the classification of these unknown image sequences. Recent years, various representation types and matching algorithms have been developed under single layered approaches. Most of them adopt a sliding windows technique that classifies all possible sub-sequences. These approaches work well for the recognition of relatively simple gestures and actions with sequential characteristics such as walking, running, and jumping. However, for some complex activities with real world background, this kind of approaches do not work very well. In this case, hierarchical approaches, which we will talk later, are better choice.

Based on the model of human activities, single layered approaches can be further divided into two types of approaches: space-time approaches and sequential approaches. Space time approaches consider the video as 3D XYT where space is X-Y dimension and time T is the third dimension. This kind of approaches classify human activities by analyzing space-time volumes of given videos. The 3D XYT models will be learned and constructed from training videos. And some other 3D models will be established corresponding to unlabeled videos. Comparing the similarity of these two kinds of models, the classification of those unlabeled videos could be determined. This algorithm is similar as the template matching algorithm which we talked in the previous section. Another kind of single layer approaches, sequential approaches, consider the video as a sequence of images and interpret the human activity as a sequence of observations. As we know, a video is composed by a sequence of images. The feature extracted from each image frame describe human status. Therefore, a sequence of images will provide a sequence of human status. Such sequence will tell us which activity is occurred by computing the maximum likelihood probability(MLP) between the sequence and the activity class. Space-time approaches are straight forward approaches and are widely used in the recognition of periodic actions. The weakness of such kind of approaches is handling the speed and motion variation.

Besides the pure 3D volume representation for space time approaches, there are another two space time representation called as trajectories approaches and space time feature approaches. In trajectory approaches, an activity can be represented as trajectories in 3D dimension. As mentioned in previous section, these trajectories, obtained by tracking, represent the movement of the person. Thus, the activity can be derived by analyzing a set of trajectories. The space time trajectory approaches provide the detail analysis and results for most cases, but body parts analysis is always difficult for this kind of approaches. Instead of pure volume or pure trajectory, a set of features

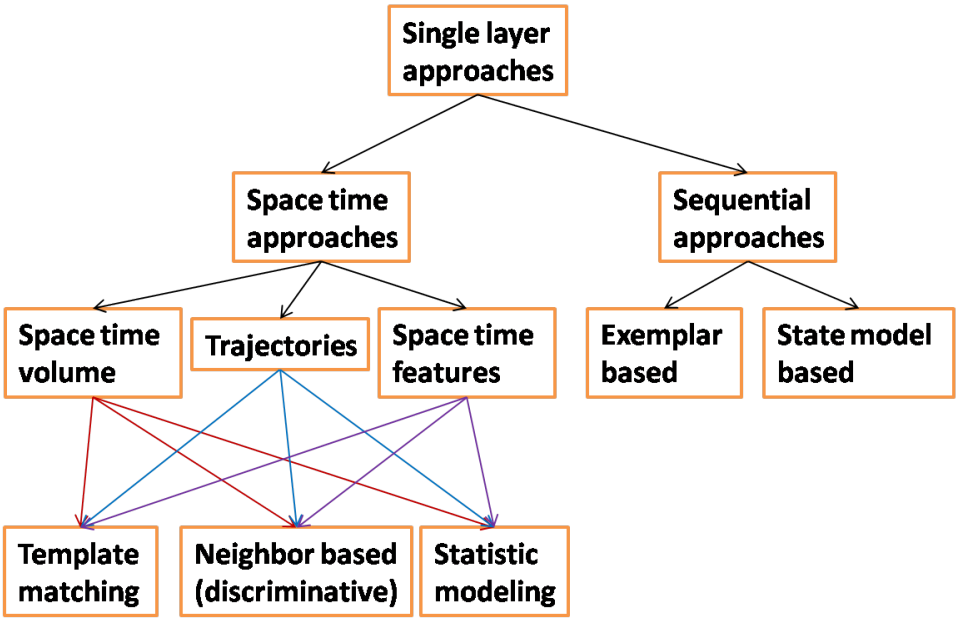

Figure 2.1: Single-layered approaches and the lists of selected publications corresponding to each category [12].

extracted from the volume or the trajectory is also used to represent human activity. In this kind of approaches, volumes or trajectories are treated as some objects where the common features can be extracted from them. This kind of approaches is reliable even under noise and illumination changes. However, the computation complexity will dramatically increase when recognizing more complex activity. In addition, viewpoint invariance has to be considered in this kind of approaches.

Space time approaches can also be categorized as in three types: template matching, neighbor-based(discriminative), and statistical modeling. In template matching approaches, the representative models for all activities are established though training videos. Comparison between these models and the models obtained from unlabeled videos will tell the classification of these unlabeled videos. In the case of neighbor-based matching, the activity was described by a set of sample volumes (or trajectories) which are used to match those obtained by the unknown input. Statistical modeling algorithms match training and testing videos by explicitly modeling a probability distribution of an activity.

For sequential approaches, we have discussed both types in the previous section pose recognition. They are exemplar based recognition and model based recognition. A tree structure taxonomy's figure of single layer approaches is shown in Fig 2.1[12].

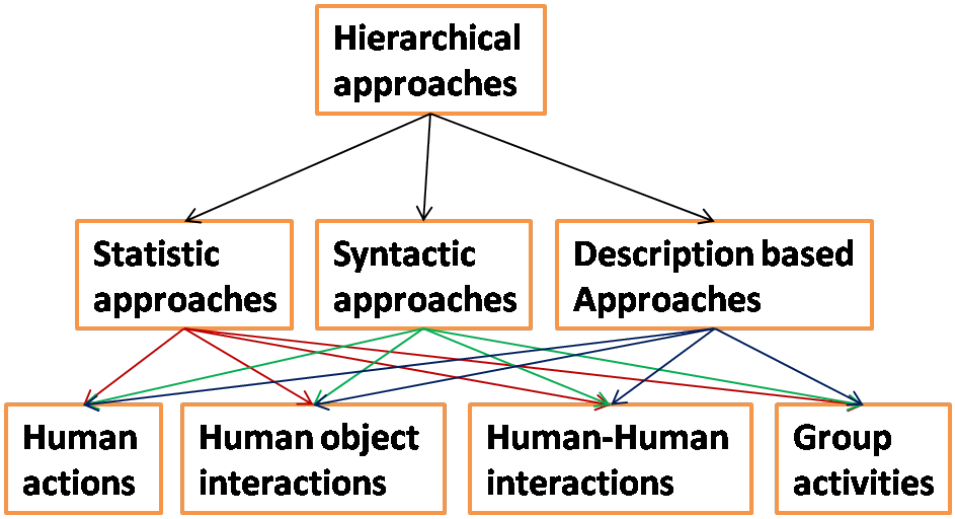

Figure 2.2: Hierachical approaches and the lists of selected publications corresponding to each category [12].

Another kind of approaches are the hierarchical approaches. It aims to recognize high-level human activity from the recognition results of other simpler activities. As we know, any complexity event is composed by multiple simpler sub-events. Therefore, the system will classify these sub-events first because they are relatively easier to be recognized, and then the higher level event can be derived from these known sub-events. Therefore, in hierarchical approaches, a high-level human activity will be decomposable until the atomic activities are obtained. The idea of hierarchical approaches greatly improving the recognition process by reducing redundancy where the recognized sub-events can be used multiple times. In addition, the layer by layer structure makes the computation traceable and easier to be understood.

As shown in Fig 2.2, hierarchical approaches can be categorized as statistical approaches, syntactic approaches, and description-based approaches. In hierarchical statistical approaches, state-based models such as Hidden Markov Model (HMMs) and Dynamical Bayesian Networks (DBNs) are used. In these models, the structure of activity recognition has multiple layers. At the bottom layer, the recognition algorithm for those atomic activities is exactly the same as that one used in single-layered approaches. Low level features are extracted from video data and are converted to a sequence of atomic activities. Then, in the second-level layer, this sequence of atomic activities is used as observations for the recognition of higher level activities. Thus, the highest level activity would be obtained following such layer by layer derivation. In each layer,

the result is calculated by computing the likelihood between the activity and the input sequence features/observation activities with the maximum likelihood estimation (MLE) or the maximum a posteriori probability (MAP) classifier. Statistical approaches have been successfully applied for the recognition of sequential activities in numbers of publications. This kind of algorithms is robust enough for activities recognition even in the case of noisy inputs. However, this kind of approaches are inherently unable to recognize activities with complex temporal structures. Therefore, their applications are limited for modeling sequential relationships instead of concurrent relationships.

As for syntactic approaches, human activities are represented as a string of symbols where each symbol corresponds to an atomic activity [22]. The same as the case of hierarchical statistical approaches, the atomic activities are recognized by those extracted low level features. These atomic activities are then parsed to symbols though provided production rules. And the high level human activities are recognized by using Context-free grammars (CFGs) and stochastic context-free grammars (SCFGs). The major limitations of syntactic approaches is also on the recognition of the concurrent activities which composed of concurrent sub-events. Besides that, another limitation comes from synthetic approaches assumption. All observations are assumed to be able to be parsed by production rules. This assumption brings troubles when an unknown observation interferes with the recognition. To overcome such limitation, some algorithms are developed for automatically learning grammar rules from observations [23].

A description-based approach represents human activities as the composition of atomic activities where the temporal, spatial and logical relationships between these atomic activities are considered. The relationship between sub-events as well as the recognition for atomic activities plays an important role for the recognition of high-level human activity. One of the advantages of the description-based approaches is that they are able to recognize those activities with concurrent structures. The limitation of description-based approaches is their inability to compensate for the failures of low-level components such as human detection failure. The recognition accuracy will be greatly reduced with these detection failures.

2.3 Complexity-based taxonomy of human activity recognition

As described in the previous section, human activity recognition can be categorized as single person action, binary interaction, and group activity based on video complexity. Single person action recognition means only one person is in such video and we classify his action into a certain action category. Since it is an element recognition of human activity, it attracts lots of attentions of researchers. Numerous algorithms were developed for both recognition methodologies and tools. Many of them are also suitable for the recognition of interactions and group activities. Since some traditional approaches are mentioned in previous section, the useful tools for activity recognition will be introduced in this section.

2.3.1 Detector and descriptor for action recognition

In computer vision, a feature detector is a tool which is used to detect the features in images or videos. A feature means a part of interest in images or videos. Human activities can be represented by features. Thus, correctly and effectively detecting features in the images or videos will greatly affect the speed and accuracy of recognition. Generally, the resulting features are in the form of isolated points, continuous curves or connected regions. For human detection, the traditional types of features are edges, corners, and blobs. Edges are some sets of points with strong gradient magnitude. Corners, also called as point of interest, are some isolated points with both strong gradient magnitude and a ”good position”. That means, these points are stable even under local or global perturbations. Blobs are connected regions. Blob detectors are similar to corner detectors but can detect those areas in an image or videos which are too smooth to be detected by a corner detector. Table 2.3 lists some common detectors for human recognition.

A Harris 3D detector detects spatial and temporal corners and provides automatic scale selection. However, ST-corners can be quite rare in an image/video. That means ST corners are too sparse for many types of motion. A cuboid detector detects regions with spatially distinguishing characteristics undergoing a complex motion. It has a rich set of features but doesn't have scale selection. A cloud ST features detector solves some problems of cuboid detector. In practice, it performs much better than a traditional cuboid detector especial in noisy environment. However, initially foreground area segmentation increases the cost of such detector. A volumetric features

name of detector	type	author and publication
Canny edge detector	Edge detector	Canny, J.,IEEE Trans. 1986
Harris3D detector	Corners detector	Laptev et al. ICCV03
Hessian detector	Corner detector	Williems et al. ECCV 2008
Cuboid detector	Corner detector	Dollr et al. ICCV 2005
Cloud ST features detecor	Corner and edge detector	Bregonzio et al. CVPR 2009
Volumetric features detector	Blob detector	Ke et al. ICCV 2005
Principal curvature-based region detector	Blob detector	Deng, H. et al. CVPR 2007

Table 2.3: Some common detectors

name	author and publication
Scale Invariant Feature Transform (SIFT)	Lowe, David G. ICCV 1999
Speeded Up Robust Features (SURF)	Bay, H et al. ECCV 2006
HOG3D descriptor	Klaser et al BMVC 2008
Optical flow descriptor	Barron, L. J. JSCV 1994
Cuboid descriptor	Dollr et al. ICCV 2005
Gradient Descriptor	Dollr et al. ICCV 2005
HOG/HOF Descriptor	Dalal N, CVPR 2005

Table 2.4: Some common descriptors

detector is a detector based on Viola and Jones rectangle features. It defines an integral video and calculated on x and y optical flow channels. This detector has dense features at many locations and scales results in efficient computation of features. But it needs to subsample the feature spaces because sometimes the features are too dense. In addition, in order to achieve spatial scale invariance, a video pyramid has to be processed. A Hessian detector is the ST extension of the Hessian saliency measure. The advantage of such detector is automatic scale selection. But examples suggest that high entropy ST-regions are rare.

Once features have been detected, extracting these features to get information of an image or video will be executed. However, the input data is often too large to be processed. To handling these redundant data, we need to transform them into a reduced representation set of features. We call this kind of representation descriptor. For example, interested points can be represented by a descriptor in an image or video. Table 2.4 lists some common descriptors.

The overall ranking for some common descriptors are: HOG/HOF $>$ HOG3D $>$ Cuboids $>$ SURF & HOG, and the combination of gradients plus optical flow also seems good choice.

Besides detector and descriptor, one other tool for human recognition is the classifier. The selection of a proper classifier will also greatly improve the recognition accuracy. k-NN is a typical instance-based prediction classifier. Based on their Euclidean distance, the classification of a testing sample will be determined by the majority class of its k closest neighbors. Naive Bayes (NB) is another classifier model. It computes the probability of classification based on the Bayes's rule. It is probably one of the most common classifiers for certain types of learning problems. Another kind of most common classifiers are Support Vector Machines (SVMs). SVMs are a kind of a blend of linear modeling and instance-based learning [24]. It separate the dataset to training samples and testing samples. A linear discriminant function which used to distinguish each class will be learned from training samples and then applied on test samples. In case there is no linear separation from training samples, SVMs kernel will make the training samples be projected into a higher-dimensional space. Then the classifier can be learned in this high dimension space. K-mean is also an important classification tool. This classifier calculates the means of initial classes which evenly distributed to whole data space. By using a minimum distance, Kmean iteratively clusters the pixels into the nearest class. In each iteration, pixels in data space are reclassified based on previous obtained means and then the class means are recalculated. This process continues until the number of pixels in each class changes by less than the selected pixel change threshold or the maximum number of iterations is reached.

Feature detector, descriptor, and classifier are not only used for the recognition of single person action, but also for the recognition of binary interaction and group activity. There are two kinds of group activity. In the first kind of group activity, all individuals' activities are similar or the same. For example, when soldiers are marching on the street, each individual soldier is walking in the same direction with same speed. Another example is queuing, people will stand on a line with similar pose. In such kind of activities, the analysis of individual action is trivial but the detection of overall motion and the group members formation are vital. Since the motion of group can be considered simultaneously, single layer approaches are good for such recognition. Through proper detector and tracker, trajectory of the group can be extracted from the video and can be compared with template for the activity analysis [25]. Additionally, each person can be treated as a point where the group can be represented as a set of points. Shape and formation changes of this set will provide sufficient clues for the recognition[26]. In another kind of group

activity, individual actions are different and each member has own role. Early researches focus on the recognition of group activity by analysis of the members with non-uniform behaviors in a single group[27, 28, 29]. For example, a teacher is doing presentation while all other students are listening in a classroom. Recent years, more challenging group activities are analyzed. In some activities, each person has different role. For such kind of group activity, the activity of each member in the scene has to be recognized and their structures should be analyzed. Therefore, the most approaches for the recognition of such group activity are hierarchical because there are at least two-levels of activities: group activity and each member activity [30, 31, 32]. The most popular approaches is statistical hierarchical approaches which has been discussed in previous section. Recent years, some methodologies have been developed handle both kinds of group activity and achieve a promising results [1, 2, 33, 34, 35].

2.3.2 Binary human-human interaction

Because of the limitation of datasets, the study of binary interaction is even behind the study of group activities. In 2000, Oliver et al propose a Bayesian model to analyze the binary interaction [36]. They obtain the trajectories of both person and compute the MLP to classify the interaction. Around 2004, J.K. Aggarwal's research group in university of Texas at Austin developed a hierarchical method for binary interaction recognition[37, 30]. They divided human motion to body part movements such as Torso's movement and arm's movement. According to head pose information and body parts information, they classified the interaction to different categories. With a new realistic dataset, this research group developed a video structure comparison method in later years [4]. This well-known new dataset is called as UT-dataset. So far, it is still the most popular dataset for binary interaction study. In their work, they extracted histogram based spatio-temporal local features from videos. After that, they create a match kernel which belong to Mercer's kernel and use this match kernel to measure the similarity of feature structures from different videos. Then they localize the detected atom activity by searching the activity's spatial coordinates, starting time, and ending time which is based on voting. Through hierarchical recognition, the detected binary interaction can be classified. With this system, more complicated binary interactions are able to be recognized. Compared with previous works, the approach proposed in their work greatly improve

the recognition accuracy for the realistic binary interaction.

With more available realistic datasets in recent couple of years, diverse methodologies were developed. One typical volumetric-based approach is proposed by Brendel et al. in 2011 [7]. They extracted pixel intensity and motion properties at multiple scales and segment them to obtain homogeneous sub-volumes, called tubes. These tubes are organized as three types of relationship: Hierarchical, Temporal, and Spatial. To simplify, they constructed a spatial-temporal graph by using nodes to represent tubes and weighted direct edges to represent these relationships. Based on these knowledge, they learned weighted least squares graph model from a set of training graphs of an activity class. Thus, the testing videos can be parsed by matching its graph with the closest activity model in the weighted least squares sense, under a arbitrary permutation. According to their results comparison tables, the performance of this approach on UT-dataset is better than that of [4].

In the same year, Guar et al. proposed another model, string of feature graphs model [8]. Different with Brendel's approach, they only divided features into small temporal bins and represented the video as a temporally ordered collection, where each feature bin consisting of a graphical structure representing the spatial arrangement of the low-level features. To match two videos, they first match these local feature bins in a graph-theoretic manner to preserve the spatial-temporal relationships between features. Then they used dynamic time wrapping for global temporal alignment. Besides binary interaction recognition, this approach is also able to recognize activities which has interactions between multiple objects. The experiments parts in their publication indicate that they achieved the comparable results with [4].

In 2012, Patron-Perez et al. developed a new approach to recognize binary interactions in video from their new TVHI dataset [5]. They tracked all upper bodies and heads in a video and developed a person centered descriptor based on the head orientations and the local spatio-temporal region around them. From the information of local cues, they obtained the spatial relationship between people and head orientations, which are called as global cues. Then they use structure SVM to learn and inference on their model to obtain the interaction class. Besides their new dataset, they also performed their model on UT dataset. The classification accuracy is even better than that of Brendel' work.

With a new BIT interaction dataset, another approach was proposed by [6]. They used high-

level descriptions, which is called interactive phrases, to represent binary semantic motion relationships between those interacting people. These motion relationships between arms, legs, and torsos could be leg stepping forward, arm stretching, static torso, and etc. And they treated these interactive phrases as latent variables. Finally, they classify the interaction types by using latent SVM. They tested their model on both BIT dataset and UT dataset and got encouraged results.

Besides above approaches, one interested approach, propagative Hough voting approach, was proposed by Yu et al. in 2012 [9]. In their work, they use propagative Hough voting to analyze the binary interaction. To start, they extracted the STIPs from videos and use random projection trees(RPT) to model the underlying low-dimension feature distribution. This leverages the low dimension manifold structure in the high dimensional feature space. By accumulating the voting score for matching features, the classification of the videos can be determined. Though this method increases some computing cost, the superior performance on UT dataset and TVHI dataset proves that it is an excellent methodology for binary interaction recognition.

Though more publication about binary interactions appear in recent years, the study of that is still not rich. In next chapter, we will propose a new recognition method for binary interaction which represent it by interaction trajectories. How to design suitable kernels for comparing interaction trajectories will be also introduced in next chapter.

Chapter 3

Framework of Binary Interaction Recognition

Recognition of binary interaction is one of the important areas for the understanding of human activity by computer. However, the explore on this area is much less than other areas of human activity recognition because of the limitation of realistic datasets. To improve the recognition accuracy for binary interaction, it is necessary to establish a modeling framework In this chapter, we will explain how to construct this framework and the principle of such framework. Compared with other frameworks, this new framework boosts both recognition performance and efficiency for binary interaction recognition [38].

In the first section of this chapter, we will describe that the representation of human interactions can be done by interaction trajectories. Besides that, we will introduce a new efficient motion feature, called motion histogram. In the second section, we will pose the human interaction recognition problem and identify the challenges it implies. In the third section, we explain kernel nonlinear dynamic system(NLDS) and how to use it to model interaction trajectories. In the fourth section, comparing interactions with kernel NLDS will be discussed.

3.1 Representation of Binary Interactions

Given a video I, we convert it to image sequences $\{I_t\}_{t=1}^{T}$, where t represents the frame number and T means the length of the sequence. For binary interaction, there should be two or more(other

Figure 3.1: An example of bounding box

people will be considered as perturbation) in the image sequences. We assume the region of each person at every frame being given though the use of tracker [39](This is a typical assumption in video surveillance setting). With this assumption, we can use the bounding box to delimit the region of each person at each frame. The features selection and extraction will be executed only in bounding box area instead of whole region of frames.

3.1.1 Histogram of Oriented Optical Flow

To effectively represent the binary interaction, we extract two kinds of features from the video. The first one is the *histogram of oriented optical flow* (HOOF) [40], $\mathbf{h}_{i,t}$. Here i means the i-th person in the video. Optical flow, as one of methods to detect human motion, is defined as apparent visual motion and the changes of light in the scene. The second row of Figure 3.2 shows an example of optical flow image. However, optical flow detection is susceptible to the variation of scales, background noise, and the direction of movement. To overcome these problems, HOOF, based on the distribution of optical flow, was proposed by Chaudhry et al. in 2009. They binned the

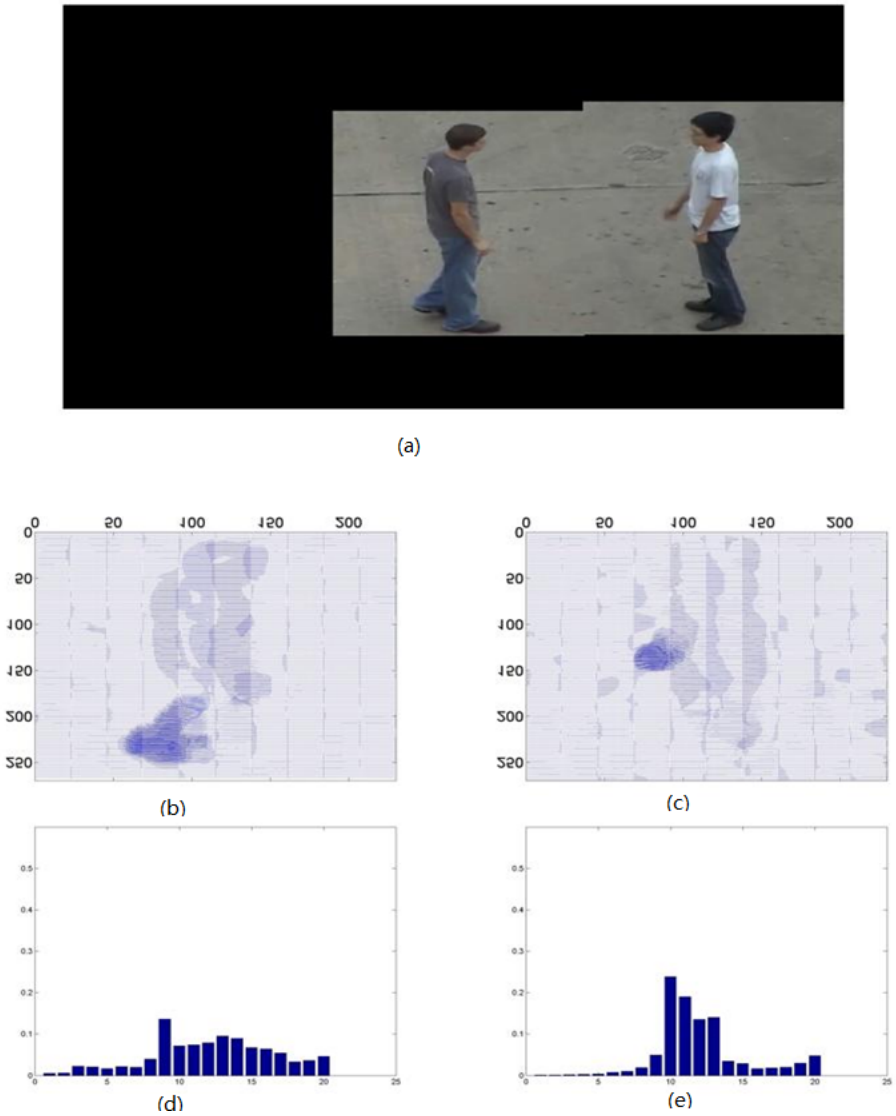

Figure 3.2: An example of HOOP descriptor. a) Binary interaction image cut from video. b) Optical flow of left person. c) Optical flow of right person. d) histogram bins obtained from b. e) histogram bins obtained from c.

flow vector though its angle and magnitude weight and then normalized the histogram. This makes HOOF be independent of direction of motion and scale variation. The third row of Figure 3.2 shows the histogram bins obtained from the optical flow images, and Figure 3.3 shows how histogram was formed in this method. From Figure 3.3, HOOF is symmtry in the orientation of the optical flow which indicates this feature is independent of direction of motion. Although, HOOF features can not be used to represent the relative direction of motion between pair persons, it prepresent each

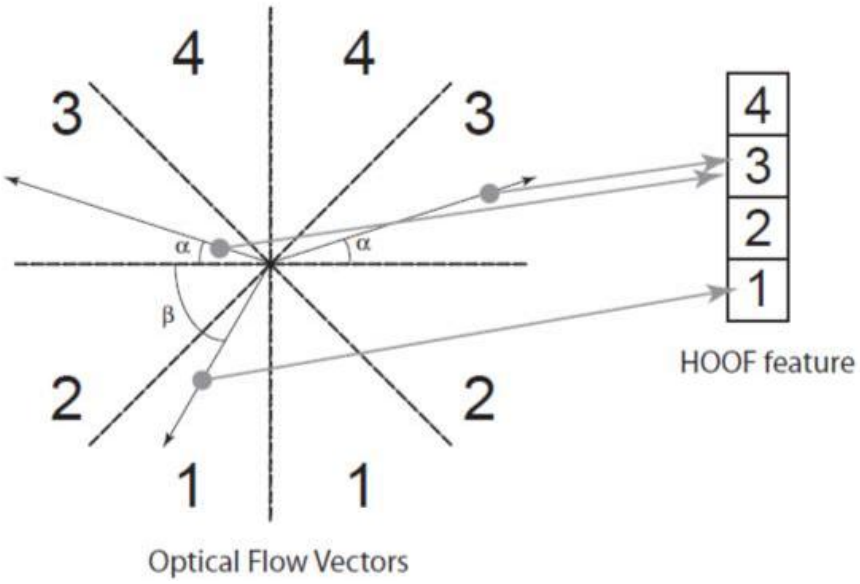

Figure 3.3: Histogram formation with 4 bins [40].

single person's motion very well. Thus, in our framework, HOOF features were used to represent the motion of each person between two consecutive frames. The relative direction of motions between two persons will be represented by another feature.

3.1.2 Motion histograms

Another kind of features we used in this framework is called as *motion histogram* (MH), which summarizes the motion trajectory of the past $\tau - 1$ frames (where $\tau > 1$). To obtain MH, we fist need to compute the *motion image*, $M_t \doteq \sum_{k=1}^{\tau-1} \eta(I_t - I_{t-k})$, where $\eta(z) = 1$ if $|z| < \delta$, otherwise $\eta(z) = 0$. Here δ is a threshold parameter to be set. Once the motion image is computed, it was used to bin inside the bounding box of person to obtain the motion histogram of person i at frame t, $\mathbf{m}_{i,t}$. Same properties as HOOF, this MH features are also scale invariant, robust to noise, and independent of direction. Figure 3.4 shows a couple of examples of motion images with the corresponding MH features. Here, vertical axis is normalized histogram and horizontal axis is the number of bins.

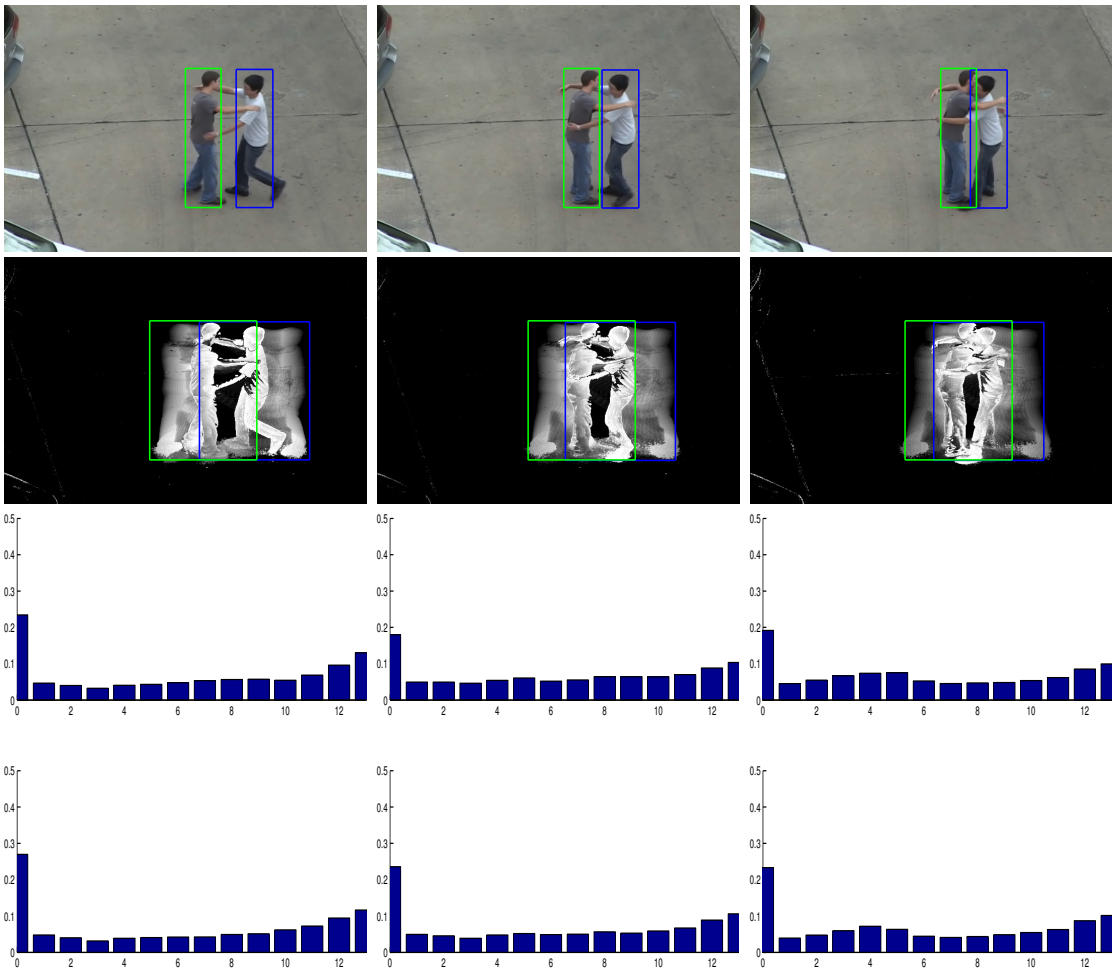

Figure 3.4: Motion images and MH feature trajectories [4]. First row: Binary interaction images obtained from video; Second row: Motion images; Third row: Motion histogram bins of left person; Fourth row: Motion histogram bins of right person.

After getting HOOF features and MH features, we use them to represent the person in the scene. For i-th person, it can be represented by the sequence of HOOF and MH features $\mathbf{h}_t \doteq \{\mathbf{h}_{i,t}\}_{t=1}^{T}$, and $\mathbf{m}_i \doteq \{\mathbf{m}_{i,t}\}_{t=1}^{T}$, respectively, where $\mathbf{h}_{i,t}$ and $\mathbf{m}_{i,t}$ are normalized histograms made of b bins, $\mathbf{h}_{i,t} \doteq [\mathbf{h}_{i,t;1}, \cdots, \mathbf{h}_{i,t;b}]^\top$, and made of τ bins, $\mathbf{m}_{i,t} \doteq [\mathbf{m}_{i,t;0}, \mathbf{m}_{i,t;1}, \cdots, \mathbf{m}_{i,t;\tau-1}]^\top$, where bin 0 has been added to account for the case of absence of motion.

Besides the features extracted from each person, the interaction between persons are also need to be considered for this representation. Here, the spatial relationship between pair persons has to be considered (e.g., person i cannot shake hands with person j if they are far enough). Generally, the spatial relationship could be obtained by analysis of the Euclidean distance between the

position $\mathbf{p}_{i,t}$ of person i, and the position $\mathbf{p}_{j,t}$ of person j [41].

$$d_{ij,t} \doteq \|\mathbf{p}_{i,t} - \mathbf{p}_{j,t}\|_2 \ . \tag{3.1}$$

The position and velocity of each person in the scene will be easily to be obtained if the camera calibration is known and people tracking is performed on the ground-plane. However, if camera is not calibrated or the calibration is not worked on the ground plane, we have to characterize proximity by approximating the distance in each frame, and performing a normalization based on the person size. Even in such case where the view is not invariant, the experiment results for the tested datasets still show a significant improving of classification accuracy.

Relative orientation between pair person is another important cue for classification. For example, person i cannot be kissing person j if i is not facing j. Such information can be obtained by the person's body part information ot gaze direction [42]. This will also lead to view invariant features. However, so far there are no available human interaction datasets with camera calibration and gaze direction information. And extracting these body part information from video is difficult beacuse a reliable 3D articulated body tracker is required. We hope to catch these information though the use of Kinetic. It is beyond the scope of this thesis. Thus, we would like to apply these information in our future works instead of this thesis.

Given the motion, described by $(\mathbf{h}_i, \mathbf{m}_i)$ and $(\mathbf{h}_j, \mathbf{m}_j)$, of person i and j, and their spatial relationship described by $d_{ij} \doteq \{d_{ij,t}\}$, their *interaction trajectory* is the temporal sequence $\mathbf{y}_{ij} \doteq \{\mathbf{y}_{ij,t}\}_{t=1}^{T}$, where

$$\mathbf{y}_{ij,t} \doteq [\mathbf{h}_{i,t}^\top, \mathbf{m}_{i,t}^\top, \mathbf{h}_{j,t}^\top, \mathbf{m}_{j,t}^\top, d_{ij,t}]^\top \ . \tag{3.2}$$

Therefore, human interactions in the scene can be represented as interaction trajectory y_{ij}.

3.2 Modeling Temporal Sequences

In previous section, we used interaction trajectory $\mathbf{y}_{ij}$ to represent human interaction. We also mentioned that $\mathbf{y}_{ij}$ is a temporal sequence and it can be considered as a section of the realization of a stochastic process which describes the dynamics of an interaction. Therefore, the recognition of binary interaction is converted to the problem of recognizing stochastic processes. Since stochastic processes is a statistical process involving a number of random variables depending on

a variable parameter such as time, it can be modeled by dynamical system. Dynamic system is a system that changes over time according to a set of fixed rules that determine how one state of the system moves to another state. It has two components: a state vector and a function. And the entire dynamical system can be then described by a set of differential equations. According to its properties, dynamical system can be further divided into some sub-systems such as linear and non-linear system, discrete or continue systems, flow or semi-flow system, and etc. In this section, we will talked about linear dynamical system and non-linear dynamical system and model our interaction trajectories with the proper system.

3.2.1 Linear Dynamical Systems (LDSs)

As we described before, dynamic system is defined as a means of describing how one state develops into another state over the course of time. For a dynamic system, if its evolution functions are linear, it is called as linear dynamic system(LDS). Otherwise, it is called as non-linear dynamic system(NDLS). This subsection will introduce the approach of linear dynamical systems(LDSs).

Mathematically, LDS evolution functions are expressed as

$$\begin{cases} x_{t+1} = Ax_t + Bv_t \\ y_t = Cx_t + \mu + w_t \end{cases} \tag{3.3}$$

Here, x_t and x_{t+1} mean the state of LDS at time t and $t+1$, respectively. y_t is the observed output at time t. A,B,C are metric coefficients where A describes the dynamics of the state evolution, B models how the state of evolution is affected by input noise, and C transforms the state of evolution to an observation. v_t and w_t are system noise and observation noise. We assume these kinds of noise are independent and zero-mean following a certain distribution such as Gaussian. μ is the mean of the past $\tau - 1$ frames, $\{y_t\}_{t=1}^{\tau-1}$. The vector spaces of these factors are $x_t \in \mathbb{R}^n$, $v_t \in \mathbb{R}^{n_v}$, $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times n_v}$, $y_t \in \mathbb{R}^m$, $\mu \in \mathbb{R}^m$, $C \in \mathbb{R}^{m\times n}$, $w_t \in \mathbb{R}^m$. Based on these parameters, the LDS can be represented as $L(x_0, A, B, C, \mu, R))$ where x_0 is the initial state and R depends on the kernel we used for noise distribution. If the noise is stable, this equation can be further simplified as

$$\begin{cases} x_{t+1} = Ax_t + v_t \\ y_t = Cx_t + w_t \end{cases} \tag{3.4}$$

Now, LDS can be represented as $L(x_0, A, C, R))$. To solve equation 3.3 and 3.4, we have to compute these parameters of $\{L_t\}_{t=1}^{T}$ first. These parameter can be learned from the feature trajectories of those training videos. There are several approaches to estimate these parameters. One typical method is using subspace identification algorithm N4SID, which is available as a Matlab toolbox [43]. However, N4SID requires a lot of memory storage if dimensionality is large. Another typical algorithm to solve this problem is given by the closed-form sub-optimal solution proposed in [44]. In this algorithm, observed value $Y_1^{\tau} \doteq [y_1, ...y_{\tau}]$ is decomposed to $U\Sigma V^T$ which called as singular value decomposition(SVD)[45] with $\Sigma = diag\{\sigma_1, ..., \sigma_n\}$. Therefore, the unique solution of best estimation is $\hat{C}(\tau) = U$ and $\hat{X}(\tau) = \Sigma V^T$. $\hat{A}$ can be determined uniquely by solving

$$\hat{A}(\tau) = \Sigma V^T D_1 V (V^T D_2 V)^{-1} \Sigma^{-1} \tag{3.5}$$

where $D_1 = \begin{bmatrix} 0 & 0 \\ I_{\tau-1} & 0 \end{bmatrix}$ and $D_2 = \begin{bmatrix} I_{\tau-1} & 0 \\ 0 & 0 \end{bmatrix}$. $\hat{B}$ is determined by input noise covariance Q by $\hat{B}\hat{B}^T = \hat{Q}$. More detailed derivation and implementation of this algorithm is given by [44]. After these parameters are determined, similarity between different LDSs will be defined through kernels such as Binet-Causchy kernel, RBF kernel, String kernel, and etc. Based on a specific kernel, all similarities of training data will be computed and used for testing data classification though support vector machine(SVM). All these algorithms to solve linear observation functions can be classified as the approach of principle component analysis (PCA).

3.2.2 Kernel Non-Linear Dynamical Systems (NLDSs)

So far, we described the LDSs approach and the algorithm for parameters estimation. However, if the point in the feature space doesn't move smoothly with time, the dynamical system is non linear. the methodology to extend LDSs to NLDSs has been proposed in a few publications [40, 46]. Instead of using PCA to learn a linear observation function in LDSs, they use kernel principle

Algorithm 1 Learning a kernel dynamic texture

Input: Video sequence $[y_1, \ldots, y_N]$, state space dimension n, kernel function $k(y_1, y_2)$.
Compute the mean: $\bar{y} = \frac{1}{N}\sum_{i=t}^{N} y_t$.
Subtract the mean: $y_t \leftarrow y_t - \bar{y},\ \forall t$.
Compute the (centered) kernel matrix $[K]_{i,j} = k(y_i, y_j)$
Compute KPCA weights α from K.
$[\hat{x}_1 \cdots \hat{x_N}] = \alpha^T K$
$\hat{A} = [\hat{x}_2 \cdots \hat{x}_N][\hat{x}_1 \cdots \hat{x}_{N-1}]^\dagger$
$\hat{v}_t = \hat{x}_t - \hat{A}\hat{x}_{t-1},\ \forall t$
$\hat{Q} = \frac{1}{N-1}\sum_{t=1}^{N-1} \hat{v}_t \hat{v}_t^T$
$\hat{y}_t = C(\hat{x}_t),\ \forall t$, (e.g. minimum-norm reconstruction).
$\hat{r} = \frac{1}{mN}\sum_{t=1}^{N} \|y_t - \hat{y}_t\|^2$
Output: $\alpha, \hat{A}, \hat{Q}, \hat{r}, \bar{y}$

Figure 3.5: Learning algorithm for kernel NLDSs [46]

components analysis(KPCA) to learn a non-linear observation function. Therefore, such dynamical system is also called as kernel NDLSs. In this section, we will introduce the principle of this kernel NLDSs.

To understand kernel NDLSs, it is necessary to learn KPCA first. Kernel PCA is the kernelized version of standard PCA) [47, 46]. As we know, the data is projected into a linear principal component in standard PCA. In KPCA, the data is projected onto the non-linear subspace and those non-linear principle components are expressed by kernel function. That means KPCA performs a non-linear feature transformation to the data, and then process these transformed data by standard PCA in the feature-space.In this method, the c-th component is defined by the map $\Phi(\cdot)$, and by the KPCA weight vector $\alpha_c \doteq v_c/\sqrt{\lambda_c}$, where λ_c and v_c are the c-th largest eigenvalue and eigenvector of the kernel matrix between the zero-mean data in the high-dimensional space, computed as $\tilde{K} = (I - \frac{1}{T}ee^\top)K(I - \frac{1}{T}ee^\top)$, where $e = [1, \cdots, 1]^\top \in \mathbb{R}^T$, and $[K]_{st} = K(y_s, y_t)$ (See [46] for detailed description and derivation).

Based on the knowledge of KPCA, now we are considering the extension of LDSs to NLDSs.

As we mentioned before, KPCA first transforms the data with the feature transformation $\Phi(\cdot)$ which induced by the kernel function, and then a standard PCA is used as it is in LDSs. So a observation sequences $\mathbf{y}_t$ can be transformed to $\Phi(\mathbf{y}_t)$. Therefore, the LDS equation is replaced by

$$\begin{cases} x_{t+1} = Ax_t + v_t \\ \Phi(\mathbf{y}_t) = Cx_t + w_t \end{cases} \tag{3.6}$$

Compared with equation 3.4, $\Phi(\cdot)$ is not necessarily to be known in equation 3.6. Thus, we can not estimate the parameters as in LDS. Moreover, because mapped space $\mathcal{H}$ could be an infinite dimension space, C should be a linear operator instead of a matrix where $C : \mathbb{R}^n \to \mathcal{H}$. To solve equation 3.6, we need to identify the parameter A, the sequence x_t, and some representation for C based on the knowledge of kernel K where $K = \{k_i\}_{i=1}^{T}$. The learning algorithm was given by [46] which is shown in Figure 3.5.

3.2.3 Stability of LDSs and NLDSs

As described in this section, an interaction trajectory is modeled as the output of a dynamical systems. Thus, it is necessary to explore the stability of dynamical systems. For example, in the case of synthesis, the estimated system should be stable because an unstable system would synthesize exploding outputs corresponding to image intensities outside of the visible range.

As for linear dynamical system with discrete time, the system is proved to be stable if all the eigenvalues of the A matrix are within the unit circle of the complex plane [48]. Since the typical data that we examine in human activity analysis does not exhibit an "exploding" trend, we can practically assume that the associated dynamical system is stable. There are also approaches to address the exceptions by replacing A with the estimation of matrix $\hat{A}$ that ensure the stability of the system [48]. Thus, the stability of the LDSs model for our binary human-human interaction problem is ensured.

For the model of NDLSs, we applied the KPCA step but then everything is linear and it doesn't change anything for the matrix A. So the stability of NDLSs can also be easily ensured.

3.3 Challenges in Modeling Binary Human Interactions

To modeling the binary interaction, we have to solve two problems. This subsection will state this couple of unique challenges.

- The measurements of the interaction trajectories $\mathbf{y}_{ij,t}$ do not live in an Euclidean space

  As we mentioned in previous sections, the interaction trajectories are construed by HOOF, MH, and proximity distance. Therefore, $\mathbf{y}_{ij,t}$ does not assumes values in an Euclidean space but in a Riemannian manifold with a nontrivial structure, which is $\mathbb{H}_b \times \mathbb{H}_\tau \times \mathbb{H}_b \times \mathbb{H}_\tau \times \mathbb{R}_+$. In particular, $\mathbb{H}_b$ is the space of histograms, which are probability mass functions satisfying the constraints $\sum_{k=1}^{b} h_{t;k} = 1$, and $h_{t;k} \geq 0$, $\forall i \in \{1, \cdots, b\}$. Thus, the interaction trajectories $\mathbf{y}_{ij,t}$ do not live in an Euclidean space.

- The decision function is expected to be symmetrical and should not be affected by any person ordering

  This is relates to the symmetry of the input feature space, which is peculiar to modeling interactions. In particular, a recognition schema entails the definition of a decision function $f : \mathbb{H}_b \times \mathbb{H}_\tau \times \mathbb{H}_b \times \mathbb{H}_\tau \times \mathbb{R}_+ \to \mathbb{R}$, which will predict whether person i and j are engaging in a certain interaction (i.e., $f(h_i, m_i, h_j, m_j, d_{ij}) > 0$), or not (i.e., $f(h_i, m_i, h_j, m_j, d_{ij}) < 0$). Therefore, given that no person ordering is imposed a priori, the decision function is expected to be symmetric with respect to i and j, i.e.,

$$f(\mathbf{h}_i, \mathbf{m}_i, \mathbf{h}_j, \mathbf{m}_j, d_{ij}) = f(\mathbf{h}_j, \mathbf{m}_j, \mathbf{h}_i, \mathbf{m}_i, d_{ji}) \ . \tag{3.7}$$

3.4 Modeling Binary Human Interaction with NLDS

As talked in previous sections, binary human interaction is represented by interaction trajectories in our project. Obviously, according to our representation, the input space of the interaction trajectory is not a linear space. Therefore, LDSs modeling is suboptimal. To effectively modeling binary human interaction which address the first challenge proposed in the previous section, we will use NDLSs.

Given a video I, we use interaction trajectory $\mathbf{y}_{ij,t}$ to present this video. Since $\mathbf{y}_t$ is temporal sequence and can be modeled by NDLS, we map it to $\Phi(\mathbf{y}_t)$, which was shown in equation 3.6. According to KPCA, given α and $\tilde{K}$ which are described in 3.2.2, the sequence of hidden states $x = [x_0; x_1; \cdots; x_{N-1}]$ and the state-transition matrix, A, can be estimated as

$$x = \alpha^\top \tilde{K} \tag{3.8}$$

$$A = [x_0; x_1; \cdots; x_{N-1}][x_0; x_1; \cdots; x_{N-2}]^\top \tag{3.9}$$

Our goal is to use NLDS for recognition, so we do not need to estimate every parameter such as C. The only thing we need to do is finding a proper method to comparing two systems. This can be done by comparing the corresponding kernel matrices $\tilde{K}$ which will be described next section.

3.5 Comparing Interactions with Kernels NLDSs

To classify human activity, we need to compare the similarity of interaction trajectories between training video and test video. That means, a method to compare similarity has to be developed. Here, we take advantage with kernel NLDSs where the similarity could be computed without knowing the map Φ and the parameter C. In our project, the kernel we used for interactions comparison is called as NLDS Binet-Cauchy kernel, which was proposed recently [40]. In particular, the Binet-Cauchy trace kernel for NLDS is the expected value of an infinite series of weighted inner products between the outputs after embedding them into the high-dimensional (possibly infinite) space using the map $\Phi(\cdot)$. More precisely

$$K_{NLDS}(\{y_t\}_{t=1}^{\infty}, \{y'_t\}_{t=1}^{\infty}) \doteq E\left[\sum_{t=1}^{\infty} \lambda^t \Phi(y_t)^\top \Phi(y'_t)\right] = E\left[\sum_{t=1}^{\infty} \lambda^t K(y_t, y'_t)\right] , \tag{3.10}$$

The same as in LDS confinement, $0 < \lambda < 1$, and the expectation of the infinite sum of the inner products is taken w.r.t. the joint probability distribution of v_t and w_t. The kernel (3.10) can be computed in closed form, and it requires the computation of the infinite sum

$$P = \sum_{t=1}^{\infty} \lambda^t (A^T)^\top F A'^\top , \tag{3.11}$$

where $C^\top C'$ is replaced by F because it can not be computed directly in NLDS environment. Now, $F = \tilde{\alpha} S \tilde{\alpha}'$, and the columns of $\tilde{\alpha}$ and $\tilde{\alpha}'$ are the centered KPCA weight vectors of $\{y_t\}$

and $\{y'_t\}$, given by $\tilde{\alpha}_c = \alpha_c - \frac{e^\top \alpha_c}{T} e$, and $\tilde{\alpha}'_d = \alpha'_d - \frac{e^\top \alpha'_d}{T'} e$, respectively. S instead is such that $[S]_{st} = K(y_s, y'_t)$, where $s \in \{1, \cdots, T\}$, and $t \in \{1, \cdots, T'\}$. Follow the same procedure in LDS, P can be computed by solving the corresponding Sylvester equation $P = \lambda A^\top P A' + F$.

Given P, kernel (3.10) can be computed in closed form provided that the covariances of the system noise, the observation noise, and the initial state are available. On the other hand, like [40] points out, for recognition of phenomena that are assumed to be made by one or multiple cycles of a temporal sequence, we want to use a kernel that is independent from the initial state and the noise processes. Therefore, the original kernel (3.10) is simplified to K^σ_{NLDS}, which is a kernel only on the dynamics of the NLDS, and is given by the maximum singular value of P, i.e.,

$$K^\sigma_{NLDS} = \max \sigma(P) \ . \tag{3.12}$$

For more details about the estimation of the NLDS model parameters, and about the derivation of kernel (3.12) the reader is referred to [44, 46, 40].

Chapter 4

Pairwise Kernel Design

In the previous chapter, we establish a framework for modeling binary interaction based on the interaction trajectories. In this frame work, the similarity between two videos can be compared through kernel. Therefore, the performance of such framework greatly depends on how well the kernel is designed. In this chapter, we will introduce the kernel for dynamical system and propose a few strategies to design the kernel K for binary interaction recognition.

4.1 Kernel for Dynamical System

For dynamical system, the similarity measure on a high dimensional space can be computed by simply computing the kernel function on the original representation. In this section, we will briefly describe some popular kernel measures used on the space of histograms or other non Euclidean spaces.

Mercer kernels, proposed in [47], are positive definite kernels that induce an inner product in a higher dimensional space, called as a Reproducing Kernel Hilbert Space(RKHS). For points lying on the non-linear manifold, the Mercer kernel is given by

$$k(h_1, h_2) = \Phi(h_1)^\top \Phi(h_2) \tag{4.1}$$

There are several representations for Mercer kernels. If we represent histogram as square root, we have $\sqrt{h_t} = [\sqrt{h_{t;1}}, ..., \sqrt{h_{t;N}}]$. Such histogram can be projected to N dimension hypersphere where the Riemannian metric between two points R_1 and R_2 on the hypersphere is $d(R_1, R_2)$. At

this case, the kernel between two histograms can be represented as

$$k_S(h_1, h_2) = \sum_{i=1}^{N} \sqrt{h_{1;i} h_{2;i}} \tag{4.2}$$

This kind of Mercer kernels is known as Geodesic kernel and can be derived from radial basis function(RBF) kernel $k(h_1, h_2) = \exp(-d(h_1, h_2))$ with Bhattacharyya distance $d_B(h_1, h_2) = -ln(BC(h_1, h_2))$, where coefficient $BC(h_1, h_2) = \sum_{i=1}^{N} \sqrt{h_{1;i} h_{2;i}}$. Notice that RBF kernel depends on the selection of distance representation.

There are several ways to represent this distance. Bhattacharyya distance measures the similarity of two discrete or continuous probability distributions [40]. Another kind of distance to measure the similarity of histogram is called as Minimum Difference of Pairwise Assignment [49] where

$$d_{MDPA}(h_1, h_2) = \sum_{i=1}^{N} \left| \sum_{j=1}^{i} (h_{1;i} - h_{2;i}) \right| \tag{4.3}$$

The third popular distance between two histograms is χ^2 distance

$$d_{\chi^2}(h_1, h_2) = \frac{1}{2} \sum_{i=1}^{N} \frac{|h_{1;i} - h_{2;i}|^2}{h_{1;i} + h_{2;i}} \tag{4.4}$$

All these kinds of distance can be used in RBF to compute the similarity of histogram.

Besides RBF kernel, another kind of Mercer kernel is called as Histogram Intersection Kernel (HIST) [50]. It is defined as

$$k_{HIST}(h_1, h_2) = \sum_{i=1}^{N} min(h_{1;i}, h_{2;i}) \tag{4.5}$$

For LDSs, a family of Binet-Cauchy kernels [51] are defined as

$$K_{LDS}(\{y_t\}_{t=1}^{\infty}, \{y'_t\}_{t=1}^{\infty}) \doteq E\left[\sum_{t=0}^{\infty} \lambda^t y_t^\top y'_t\right] \tag{4.6}$$

where $0 < \lambda < 1$, and the expectation of the infinite sum of the inner products is taken with regards to the joint probability distribution of v_t and w_t. Assuming underlying and independent noise processes are the same, the trace kernel of Binet-Cauchy could be obtained in close form

$$K_{LDS}(\{y_t\}_{t=1}^{\infty}, \{y'_t\}_{t=1}^{\infty}) = x_0^\top P x_0{}' + \frac{\lambda}{1-\lambda} trace(QP + R) \tag{4.7}$$

here, Q and R are covariances for the state and output, respectively. And P is represented as

$$P = \sum_{t=0}^{\infty} \lambda^t (A^t)^\top C^\top C' A'^t \tag{4.8}$$

and can be further solving if $\lambda||A||||A'|| < 1$ [40, 51]

$$P = \lambda A^\top P A' + C^\top C' \tag{4.9}$$

$||.||$ is a matrix norm and (i, j)-th entry of $C^\top C'$ is $c_i^\top c'_j$ where c_i and c_j are the i-th and j-th principal components respectively.

4.2 Pairwise Kernel Design

In our framework model, the input feature space $\mathcal{S} \doteq \mathbb{H}_b \times \mathbb{H}_\tau \times \mathbb{H}_b \times \mathbb{H}_\tau \times \mathbb{R}_+$ is a non-Euclidean space which is a Riemannian manifold. Therefore, the kernel K should be defined as a non-linear term. There are several ways to construct a non-linear kernel. One way is to extend a linear kernel such as RBF kernel with Euclidean distance to a non-linear kernel with non-Euclidean distance. In order to take advantage of the known Riemannian structure of $\mathcal{S}$, we have to replace the Euclidean distance by the distance for the manifold $\mathcal{S}$ in the kernel design. However, we have not found any good method to define distance on manifold space $\mathcal{S}$. It brings problem for our kernel design. Alternative approach for this problem is using kernel construction techniques which are discussed in [47]. In this reference, $\mathcal{S}$ is represented by the Cartesian product of subspaces. Therefore, we can concentrate on each subspace separately and exploit the known geometry to the full extent.

Now, we design a histogram kernel K_H and a distance kernel K_d where K_d means the distance between two people. Since the input feature space $\mathcal{S}$ is represented by the Cartesian product of the subspaces, we compute K_H in the first subspace $\mathbb{H}_b \times \mathbb{H}_\tau \times \mathbb{H}_b \times \mathbb{H}_\tau$, and K_d in the second subspace $\mathbb{R}_+$. Follow the method proposed in [47], a tensor product kernel K can be computed though the combination of K_H and K_d which is expressed as

$$K \doteq (K_H \otimes K_d)(y_{ij}, y'_{ij}) = K_H((h_i, m_i, h_j, m_j), (h'_i, m'_i, h'_j, m'_j)) K_d(d_{ij}, d'_{ij}) \ , \tag{4.10}$$

To lighten the notation, the time subscript t is not shown in above equation. According to the equation 4.10, a kernel K's value depends on both K_H and K_d. That means,only if the instances

in each subspace have high similarity with the corresponding instances in the same subspace, the similarity in an input space will be high. With such kernel, the classification of binary interactions was decided not only by the similarity of motion features but also by the similarity of proximity cues, as it is explained in previous chapter.

Now, let's consider K_H and K_d separately. From equation 4.10, K_d depends on the distance between i-th person and j-th person d_{ij} where d_{ij} belongs to $\mathbb{R}_+$. Since it is a Euclidean distance space, RBF is a suitable kernel here. Thus, we have

$$K_d(d_{ij}, d'_{ij}) \doteq \exp(-\gamma|d_{ij} - d'_{ij}|^2) \ . \tag{4.11}$$

Kernel K_H here depends on HOOF features and motion features. We called it as pairwise kernel [52] because the kernel $K_H : (\mathcal{X}_H \times \mathcal{X}_H) \times (\mathcal{X}_H \times \mathcal{X}_H) \to \mathbb{R}$, where $\mathcal{X}_H \doteq \mathbb{H}_b \times \mathbb{H}_\tau$, could be used to support pairwise classification. That means this kernel can be used to determine whether the examples of a pair $(a, b) \in \mathcal{X}_H \times \mathcal{X}_H$ belong to the same class or not.

In previous chapter, we also talked about the decision function which is expected to be symmetric with respect to i-th person and j-th person. The designed kernel is also expected to have such symmetry property. Thus, K_H is required to be positive semidefinite where

$$K_H((a, b), (a', b')) = K_H((a', b'), (a, b)) \ , \tag{4.12}$$

for all $a, b, a', b' \in \mathcal{X}_H$. Furthermore, by using kernel construction techniques based on direct sum and tensor product of kernels, given the kernel $k_H : \mathcal{X}_H \times \mathcal{X}_H \to \mathbb{R}$, one can build the following pairwise versions of K_H

$$K_H^D = (k_H \oplus k_H)(a, b, a', b') = k_H(a, a') + k_H(b, b') \ , \tag{4.13}$$

$$K_H^T = (k_H \otimes k_H)(a, b, a', b') = k_H(a, a')k_H(b, b') \ , \tag{4.14}$$

which obviously satisfy the symmetric property. Now we are considering to use equation 4.13 and equation 4.14 to construct a decision function f for interaction trajectories where f satisfy the symmetry property. To do so, we use a SVM to learn decision functions f based on the general kernel equation 3.10. Therefore, the decision function f will be assumed as the form

$$f(\{a_{i,t}, a_{j,t}, d_{ij,t}\}) \doteq \sum_{u,v} \alpha_{uv}\ell_{uv}K_{NLDS}(\{a_{i,t}, a_{j,t}, d_{ij,t}\}, \{a'_{u,t}, a'_{v,t}, d'_{uv,t}\}) + \beta \ , \tag{4.15}$$

In this equation, α_{uv}, ℓ_{uv}, and β are the usual SVM parameters [47], and $a_{i,t} = (h_{i,t}, m_{i,t}) \in \mathcal{X}_H$, and $a_{j,t} = (h_{j,t}, m_{j,t}) \in \mathcal{X}_H$. More importantly, equation 4.15 indicates that the symmetry property of 3.7 should be expressed as

$$\begin{aligned} K_{NLDS}(\{a_{i,t}, a_{j,t}, d_{ij,t}\},\{a'_{u,t}, a'_{v,t}, d'_{uv,t}\}) = \\ K_{NLDS}(\{a_{j,t}, a_{i,t}, d_{ji,t}\}, \{a'_{u,t}, a'_{v,t}, d'_{uv,t}\}) \ , \end{aligned} \tag{4.16}$$

for all $a_{i,t}, a_{j,t}, a'_{u,t}, a'_{v,t} \in \mathcal{X}_H$, and $d_{ij,t}, d'_{uv,t} \in \mathbb{R}_+$. Conversely, equation 4.16 leads to a symmetry property on the kernel equation 4.10 through equation 3.10, which is given by

$$K((a_{i,t}, a_{j,t}, d_{ij,t}), (a'_{u,t}, a'_{v,t}, d'_{uv,t})) = K((a_{j,t}, a_{i,t}, d_{ji,t}), (a'_{u,t}, a'_{v,t}, d'_{uv,t})) \ , \tag{4.17}$$

Note d in equation 4.17 represent the distance between two different person. Therefore, this term is symmetric where $d_{ij,t} = d_{ji,t}$ and $d_{uv,t} = d_{vu,t}$. Equation 4.17 can be further simplified to the following relationship

$$K_H((a_{i,t}, a_{j,t}), (a'_{u,t}, a'_{v,t})) = K_H((a_{j,t}, a_{i,t}), (a'_{u,t}, a'_{v,t})) \ , \tag{4.18}$$

to be valid for all $a_{i,t}, a_{j,t}, a'_{u,t}, a'_{v,t} \in \mathcal{X}_H$. Here, the relationship equation 4.18 is different than the symmetry relationship equation 4.12, and kernels that satisfy equation 4.18 are called balanced kernels [52]. Now, let's see the symmetric pairwise kernels K_H^D, and K_H^T, which are defined in equation 4.13 and equation 4.14. Obviously, they are not balanced. To solve this problem, we test two kernels that have been proved to have good theoretical properties [52], in that they guarantee minimal loss of information, and can be thought of as the balanced versions of K_H^D, and K_H^T. These two kinds of kernel are defined as follows

$$K_H^{DS}((a,b),(a',b')) = K_H^{SD}((a,b),(a',b')) + K_H^{ML}((a,b),(a',b')) \ , \tag{4.19}$$

$$K_H^{TL}((a,b),(a',b')) = \frac{1}{2}(k_H(a,a')k_H(b,b') + k_H(a,b')k_H(b,a')) \ , \tag{4.20}$$

where

$$K_H^{SD}((a,b),(a',b')) = \frac{1}{2}(k_H(a,a') + k_H(a,b') + k_H(b,a') + k_H(b,b')) \ , \tag{4.21}$$

$$K_H^{ML}((a,b),(a',b')) = \frac{1}{4}(k_H(a,a') - k_H(a,b') - k_H(b,a') + k_H(b,b'))^2 \ . \tag{4.22}$$

Here, K_H^{TL} is called tensor learning pairwise kernel [53], and K_H^{DS} is called direct sum pairwise kernel [52].

The last step is to design k_H. As we described in previous chapter, k_H is defined on the space $(\mathbb{H}_b \times \mathbb{H}_\tau) \times (\mathbb{H}_b \times \mathbb{H}_\tau)$. Since it is not required to be balanced, and both features, $h_{i,t}$ and $m_{i,t}$, should be used at the same time towards establishing similarity, we apply the tensor product rule to further decompose k_H into two kernels, $k_h : \mathbb{H}_b \times \mathbb{H}_b \to \mathbb{R}$ and $k_m : \mathbb{H}_\tau \times \mathbb{H}_\tau \to \mathbb{R}$, producing

$$k_H((h_{i,t}, m_{i,t}), (h'_{i,t}, m'_{i,t})) = k_h(h_{i,t}, h'_{i,t}) k_m(m_{i,t}, m'_{i,t}) \ . \tag{4.23}$$

Both k_h and k_m are kernels for comparing histograms. There are several options in this domain, as it is outlined in [40], where it has been shown that an excellent compromise between performance and speed is given by the geodesic kernel, which is derived by taking into account that $\mathbb{H}_b$ is diffeomorphic with the hypersphere $\mathbb{S}^{b-1}$. Both k_h and k_m are picked to be geodesic kernels for histograms with b and τ bins, respectively.

Chapter 5

Experimental Results

In this chapter, we will talk about the dataset we used in our project. In addition, some experiemental results by using different kernels were shown and the best kernel for binary human interactions is selected. Results comparison with recent publications is also performed.

5.1 Dataset

The datasets we have for our experiment include three state-of-the-art human interaction datasets. The first is the UT-Interaction dataset [4]. It contains videos of six interaction classes: *hand shake*, *hug*, *kick*, *point*, *punch*, and *push*. Since we are only interested in binary interaction, the single person action class *point* is not included in our experiment. The whole dataset is divided into two parts: Set 1 ans Set 2. The difference between them is that Set 1 videos have mostly a static background while Set 2 videos have some background motion, with some small camera motion. Therefore, it should be slightly more challenging for binary interaction recognition from Set 2 than that from Set 1. Each class in both Set 1 and Set 2 consists of 10 videos.

The second dataset is TVHI dataset. It has videos from 5 different classes: *hand-shakes*, *high-fives*, *hugs*, *kisses*, and *negative* examples. There are a large length variation (from 30 to 600 frames) and a great degree of variation among the videos as they are compiled from different TV shows, which makes this dataset very challenging. As people tracking information we were able to use the ground-truth annotations made available along with the videos, consisting of bounding boxes framing the upper bodies of all the actors in the scene. Our analysis was limited to the

bounding boxes corresponding to the people interacting, and the features were extracted from boxes having a width that was double the original annotations, in order to analyze the motion in a region surrounding each person. Note that, similarly to [5], some of the original videos where not considered due to their very limited length, or due to sharp view point changes during the interaction.

The last dataset is BIT dataset [6] which has 8 classes of human interactions: *bow*, *boxing*, *handshake*, *high-ve*, *hug*, *kick*, *pat*, and *push*. Each class includes 50 videos captured from realistic scene. Same as other two datasets, all people in the scene are annotated by bounding box. For each class, there are big variations of people's appearance, scales, illumination condition, background, and view points. Moreover, interaction people is occluded or partial occluded by other people or objects in many videos. Therefore, the recognition of binary interaction on this dataset is more challenge than that on UT-dataset. For the purpose of comparison, we use random 272 videos as training samples and 128 videos as testing which is exactly the same setting as [6].

To process these dataset, we have to detect and track the people in the scene. Low quality detectors and trackers will lead to bad feature extractions which result in degradation of performance. For example, if the tracks are fragmented, the approach brakes at the moment. However, analyzing this aspect is beyond the scope of this thesis and will be the subject of future works. Therefore, as pointed in Chapter 1, we assume that tracking information is available to simply the experiments. That means, the tracking information is simulated by annotating the data. This is a common assumption in human activity analysis. Considering all our three datasets, TVHI dataset already has the annotation information. Therefore, we annotate other two datasets manually with the VATIC tool [54] in our experiments. Figure 3.2 and Figure 3.4 give examples of how we process these datasets. We use bounding boxes to tightly bound each people in the scene at each frame. To compute the MH and the HOOF features, we place the same boxes with a width that is three times of the original to each frame. This process is shown in the second row of Figure 3.4. Here, the motion images are computed with respect to the L channel of the Lab color space, and the HOOF features are based on the optical flow computed in C++ with the OpenCV library. Another feature we obtained from proximity cues. Here, these proximity cues are limited to be the distance. Since no calibration information are provided for all these datasets, we are unable to get ground truth information. In our experiments, we normalized the distance with respect to the mean height of

the two individuals participating in the interaction.

5.2 Experiments

In our experiments, we tested the influence of recognition accuracy by different kernel construction which are proposed in previous section. Several possible choices of K_H are evaluated. And for each kind of K_H, we compute the recognition accuracy at the case of interaction trajectories with or without proximity cues. Presence or absence of this information is well marked on the tables, and also on the table kernel labels, by the presence or absence of the k_d kernel equation 4.11.

Since we extracted two features, Motion feature and HOOF feature, from videos, the combined input features $(\mathbf{h}_{i,t}, \mathbf{m}_{i,t}, \mathbf{h}_{j,t}, \mathbf{m}_{j,t})$ live in a subspace of $\mathbb{H}_{2b+2\tau}$. Based on input features, it is possible to test the following choices for K_H: (a) k_S, which is the geodesic kernel equation 4.2; (b) K_H^{TL} equation 4.20, where k_H is a geodesic kernel, indicated with $K_H^{TL}(k_S)$; (c) K_H^{DS} equation 4.19, where k_H is a geodesic kernel, indicated with $K_H^{DS}(k_S)$; (d) K_H^{TL} equation 4.20, where k_H is the tensor product kernel equation 4.23, indicated with $K_H^{TL}(k_h k_m)$; (e) K_H^{DS} equation 4.19, where k_H is the tensor product kernel equation 4.23, indicated with $K_H^{DS}(k_h k_m)$. Finally, for kernel K equation 4.10 we also tested a Gaussian RBF kernel with Euclidean distance. The kernels described above were used to train the muti-class classifier of the libSVM [55].

5.2.1 Results for the UT dataset

For UT datasets, we tested the kernel's influence on classification accuracy and the impact of features selection. Table 5.1 shows the classification accuracy for the UT-Interaction dataset by using different kernels. And Table 5.2 shows for the various kernels how classification performance is affected in three cases, namely when only the MH features are used, only the HOOF features are used, and when both are used. From them, we can obtain following information: a) The selection of kernels has a huge impact on the recognition accuracy; b) Incorporating proximity information to the kernel greatly improves the recognition accuracy; c) The best classification accuracy is obtained by using tensor learning pairwise kernel $K_H^{DS}(k_h k_m)$; d) Using an RBF kernel with Euclidean distance leads to suboptimal results; e) The proposed motion histogram features are effectively able to capture valuable motion history information, which is as discriminative as

SET 1						
Kernel/Class	Hug	Kick	Push	Punch	Hand Shake	AVG
	No Proximity					
k_S	75.00	75.00	46.15	33.33	75.00	60.65
$K_H^{TL}(k_S)$	83.33	75.00	61.53	41.66	91.66	70.49
$K_H^{DS}(k_S)$	83.33	75.00	38.46	33.33	91.66	63.93
$K_H^{TL}(k_h k_m)$	83.33	83.33	84.61	8.33	100	70.49
$K_H^{DS}(k_h k_m)$	83.33	75.00	38.46	33.33	91.66	63.93
	With Proximity					
RBF	100	100	76.92	50.00	83.33	81.96
$k_S k_d$	83.33	83.33	61.53	41.66	83.33	70.49
$K_H^{TL}(k_S)k_d$	91.66	83.33	76.92	91.66	100	88.52
$K_H^{DS}(k_S)k_d$	100	83.33	69.23	50	91.66	78.68
$K_H^{TL}(k_h k_m)k_d$	100	100	69.23	91.66	100	**91.80**
$K_H^{DS}(k_h k_m)k_d$	100	83.33	69.23	66.66	91.66	81.96

SET 2						
Kernel/Class	Hug	Kick	Push	Punch	Hand Shake	AVG
	No Proximity					
k_S	72.72	36.36	37.5	8.33	87.5	50.00
$K_H^{TL}(k_S)$	54.54	54.54	62.5	8.33	87.5	56.06
$K_H^{DS}(k_S)$	54.54	54.54	25.00	8.33	93.75	48.48
$K_H^{TL}(k_h k_m)$	72.72	63.63	37.50	50.00	62.50	56.06
$K_H^{DS}(k_h k_m)$	45.45	27.27	43.75	16.16	87.50	46.96
	With Proximity					
RBF	100	45.45	87.50	41.66	81.25	72.72
$k_S k_d$	100	54.54	81.25	41.66	75.00	71.21
$K_H^{TL}(k_S)k_d$	81.81	72.72	50.00	16.16	75.00	59.09
$K_H^{DS}(k_S)k_d$	90.90	27.27	50.00	33.33	93.75	60.60
$K_H^{TL}(k_h k_m)k_d$	100	63.64	87.50	75.00	100	**86.36**
$K_H^{DS}(k_h k_m)k_d$	100	18.18	87.50	41.66	100	72.72

Table 5.1: Classification accuracy for the UT-Interaction dataset [4]. For Set 1 MH features are computed with $\tau = 14$, and $\delta = 2$; HOOF features are computed with $b = 18$; NLDS order is set to $n = 8$. For Set 2 MH features are computed with $\tau = 22$, and $\delta = 2$; HOOF features are computed with $b = 26$; NLDS order is set to $n = 14$.

SET 1			
Kernel/Feature	MH	HOOF	Both
$k_S k_d$	65.57	68.85	70.49
$K_H^{TL}(k_S)k_d$	68.85	73.77	88.52
$K_H^{DS}(k_S)k_d$	70.49	70.49	78.68
$K_H^{TL}(k_h k_m)k_d$	-	-	**91.80**
$K_H^{DS}(k_h k_m)k_d$	-	-	81.96
SET 2			
Kernel/Feature	MH	HOOF	Both
$k_S k_d$	57.58	56.06	71.21
$K_H^{TL}(k_S)k_d$	50.00	54.55	59.90
$K_H^{DS}(k_S)k_d$	53.03	54.55	60.60
$K_H^{TL}(k_h k_m)k_d$	-	-	**86.36**
$K_H^{DS}(k_h k_m)k_d$	-	-	72.72

Table 5.2: Classification accuracy for the UT-Interaction dataset obtained using proximity and different motion features, including only motion histograms (MH), only HOOF features, and both. Motion features are computed as indicated in Table 5.1.

the information captured by the HOOF, and also orthogonal to it, given the significant boost in classification accuracy.

According to these information, we used tensor learning pairwise kernel to compare the similarity of our interaction trajectories. Figure 5.1 shows the confusion matrices by using such kernel. The numbers in this confusion matrices are the numbers of videos we tested.

To evaluate the performance of our framework, we compared our results to those results ob-

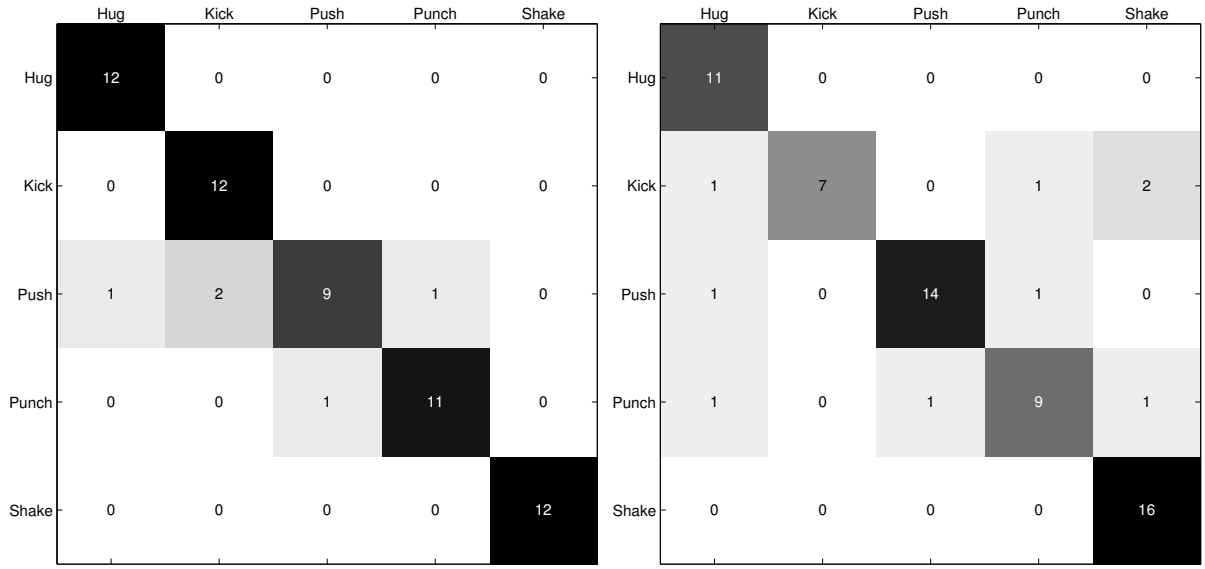

Figure 5.1: Confusion matrices for the UT-Interaction dataset: Set 1 (left), and Set 2 (right).

tained from state-of-the-art methods. Table 5.3 shows the comparison by using set1 and set2 separately, and Table 5.4 shows the comparison by using whole dataset with recent publications. From the comparison tables, we concluded that the results of our method are better or comparable with the results from recent publications.

Besides improving the performance, our method also reduce the training time. For binary human-human interaction, the actions are not symmetric in most cases. Therefore, we have to create a symmetric datasets by double the asymmetric datasets. However, in our method, we takes advantages of pairwise and balanced properties of kernels which impose the decision functions are insensitive with respect to the people ordering. So the dataset do not need to be symmetric for training, and this reduce the training time by a factor of four. In addition, the dynamical systems are good for real-rime recognition which can be learned online. It brings the possibility to use our interaction recognition approach in an online fashion.

5.2.2 Results for the TVHI dataset

The same as previous, we first tested the kernel's impact on the classification accuracy. The left side of Table 5.5 shows the classification results for the TVHI dataset.

For this dataset we have also performed a video retrieval experiment. For example, the perfect

Methods/ Set 1 UT Dataset	Hug	Kick	Push	Punch	Hand Shake	AVG
Patron-Perez et al. [5]	100%	80%	80%	60%	100%	84%
Yu et al. [9]	100%	100%	100%	60%	100%	92%
Our method	100%	100%	69%	92%	100%	92%
Methods/ Set 2 UT Dataset						
Patron-Perez et al. [5]	90%	90%	90%	70%	90%	86%
Yu et al. [9]	90%	100%	100%	100%	70%	93%
Our method	100%	73%	88%	75%	100%	88%

Table 5.3: Recognition Accuracy comparison on the UT interaction Dataset Set 1 and Set 2

Methods	Hug	Kick	Push	Punch	Hand Shake	AVG
Patron-Perez et al. [5]	95%	95%	85%	65%	85%	85%
Brendel et al. [7]	90%	69%	83%	85%	82%	82%
Gaur et al. [8]	76%	71%	79%	62%	76%	73%
Kong et al. [6]	80%	100%	90%	90%	80%	88%
Yu et al. [9]	95%	100%	100%	80%	85%	92%
Ryoo et al. [4]	88%	75%	75%	50%	75%	73%
Our method	100%	86%	79%	84%	100%	90%

Table 5.4: Recognition Accuracy of methods on whole UT interaction Dataset

retrieval score of handshake should equal to 1 when all the handshake clips in the test data are retrieved first. In particular, we have converted the proposed kernels in pairwise distances, where the kernel K_{NLDS} is normalized to 1 when $\{y_t\} = \{y'_t\}$, by computing

$$\tilde{K}(\{y_t\}, \{y'_t\}) \doteq K_{NLDS}(\{y_t\}, \{y'_t\}) / \sqrt{K_{NLDS}(\{y_t\}, \{y_t\}) K_{NLDS}(\{y'_t\}, \{y'_t\})} \tag{5.1}$$

and the distance between two interaction trajectories becomes

$$d(\{y_t\}, \{y'_t\}) \doteq 2(1 - \tilde{K}(\{y_t\}, \{y'_t\}) \tag{5.2}$$

The right side of Table 5.5 shows the retrieval precision, as defined in [56]. From Table 5.5 table, we further proved those information obtained from UT-dataset. Now, we can draw a number of considerations. First, as pointed out in previous chapter, using an RBF kernel with Euclidean distance leads to suboptimal results. Second, we have experienced a higher degree of good performance consistency for the tensor learning pairwise kernel $K_H^{DS}(k_h k_m)$, versus the direct sum pairwise kernel $K_H^{DS}(k_h k_m)$. Third, we have verified the importance of designing kernels by taking into account the structure of the input feature space in the way that different kernels rank in terms

CLASSIFICATION ACCURACY						
Kernel/Class	HS	HF	HG	KS	NG	AVG
	No Proximity					
k_S	33.33	51.72	36.36	30.43	52.00	40.77
$K_H^{TL}(k_S)$	19.05	58.62	18.18	47.83	60.00	40.74
$K_H^{DS}(k_S)$	0	62.07	31.82	30.43	88.00	42.46
$K_H^{TL}(k_h k_m)$	38.10	44.83	31.82	30.43	44.00	37.84
$K_H^{DS}(k_h k_m)$	9.52	41.38	31.82	43.48	68.00	38.84
	With Proximity					
RBF	4.76	65.52	59.09	73.91	60.00	52.66
$k_S k_d$	19.05	62.07	86.36	73.91	56.00	59.48
$K_H^{TL}(k_S)k_d$	19.05	79.31	81.82	65.22	64.00	61.88
$K_H^{DS}(k_S)k_d$	23.81	51.72	90.91	78.26	64.00	61.74
$K_H^{TL}(k_h k_m)k_d$	28.57	79.31	86.36	65.22	64.00	**64.69**
$K_H^{DS}(k_h k_m)k_d$	38.10	51.72	81.82	73.91	72.00	63.51

RETRIEVAL PRECISION					
Kernel/Class	HS	HF	HG	KS	AVG
	No Proximity				
k_S	0.239	0.316	0.293	0.402	0.314
$K_H^{TL}(k_S)$	0.208	0.335	0.265	0.498	0.330
$K_H^{DS}(k_S)$	0.199	0.300	0.267	0.424	0.300
$K_H^{TL}(k_h k_m)$	0.267	0.264	0.277	0.523	0.330
$K_H^{DS}(k_h k_m)$	0.222	0.296	0.263	0.422	0.302
	With Proximity				
$k_S k_d$	0.310	0.319	0.559	0.541	0.427
$K_H^{TL}(k_S)k_d$	0.334	0.333	0.485	0.482	0.412
$K_H^{DS}(k_S)k_d$	0.339	0.351	0.538	0.483	0.424
$K_H^{TL}(k_h k_m)k_d$	0.342	0.357	0.551	0.525	0.439
$K_H^{DS}(k_h k_m)k_d$	0.351	0.338	0.554	0.540	**0.440**

Table 5.5: Classification accuracy, and video retrieval average precision for the TVHI dataset [5]. MH features are computed with $\tau = 5$, and $\delta = 3$; HOOF features are computed with $b = 10$; NLDS order is set to $n = 10$.

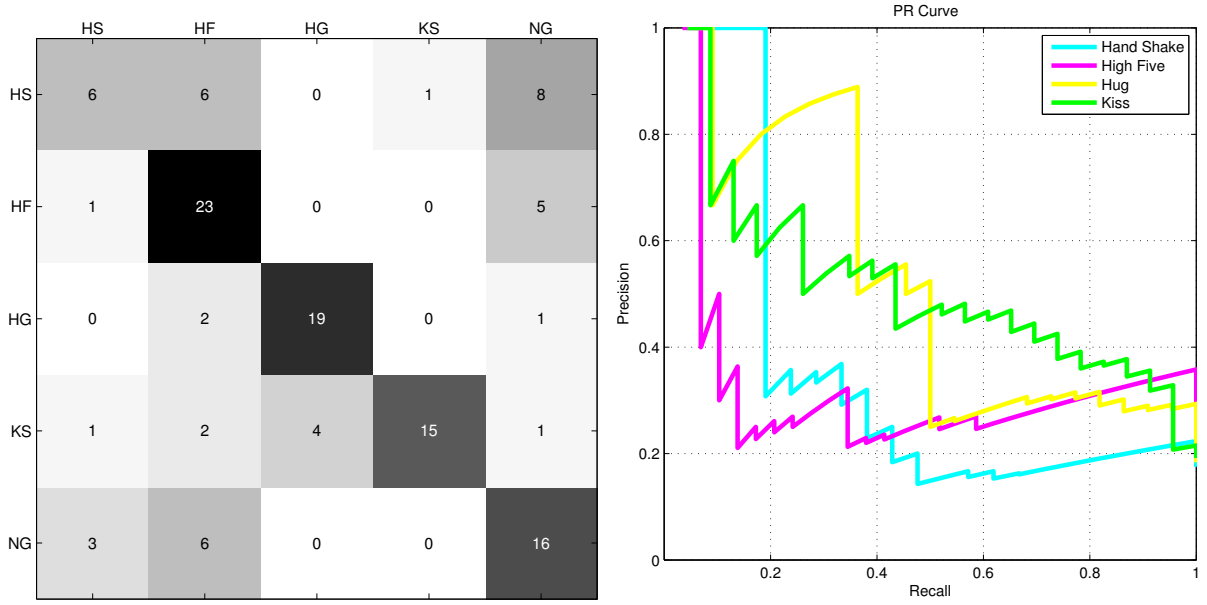

Figure 5.2: Confusion matrix (left), and per-class precision-recall curves (right) for the TVHI dataset.

of performance. Fourth, we have verified the importance in incorporating proximity information for discriminating between interactions.

We also plot the confusion matrices by using tensor learning pairwise kernel here (See the left side of Figure 5.2), and the corresponding per-class precision-recall curves [56] (See the right side of Figure 5.2).

Finally, the classification accuracy and retrieval precision by using tensor learning pairwise

Methods	Hand Shake	High Five	Hug	Kiss	Negative	AVG
Patron-Perez et al [5]	30%	62%	31%	40%	83%	49%
Our method	29%	80%	86%	65%	64%	65%

Table 5.6: Recognition Accuracy of methods on the TVHI dataset

Methods	Hand Shake	High Five	Hug	Kiss	AVG
Patron-Perez et al [5]	.39	.46	.47	.38	.42
Our method	.35	.34	.55	.54	.44

Table 5.7: Retrieval precision of methods on TVHI datasets

kernel were compared with those recently reported results. Table 5.6 and Table 5.7 show these comparisons. It can be seen that the results are comparable to the ones in [5] which shown in Table 5.6 and Table 5.7. We expect that by using the proposed kernels in a "learning to rank" approach [57], the retrieval precision would undergo a substantial increase.

5.2.3 Results for the BIT dataset

Beside the comparisons on UT-dataset and TVHI dataset, we also perform the comparison between our method and [6] on BIT interaction dataset. Table 5.8 indicates that recognition accuracy of our method is competitive.

From all of these comparisons, we can conclude that our method is on-par or better than all other recently reported results, indicating that the approach is promising.

Methods	Kong et al. [6]	our method
Bow	81%	89%
Box	81%	98%
Handshake	81%	96%
High five	94%	76%
Hug	94%	83%
Kick	81%	100%
Pat	81%	74%
Push	88%	98%
AVE	85%	89%

Table 5.8: Recognition Accuracy comparison on BIT interaction dataset. MH features are computed with $\tau = 10$, and $\delta = 10$; HOOF features are computed with $b = 11$; NLDS order is set to $n = 15$

Chapter 6

Conclusion

In this thesis, we established a framework to model binary human interactions, which is considered as the building block for human activity recognition. However, in the past years, the study of binary interaction is much less than the study of single person action and group activity because of the limitation of realistic datasets. Only until recent years, a few research group focus on the algorithm development and model construction for the binary interaction.

At start chapters, this thesis gives a general review of previous works for human activities and binary human-human interactions. Then this thesis establishes a framework to represent and recognize the binary human interaction. In this framework, an interaction trajectory, constructed as a temporal sequence, is used to represent binary human interactions in the sample videos. To compose the feature space for such interaction trajectory, this thesis proposed to use the motion histogram as a feature complementary to the HOOF feature. In addition, proximity, which is limited to distance in this thesis, was also considered. Mathematically, temporal sequences can be represented by stochastic processes, which can be modeled as the output of dynamical system. Therefore, the recognition problem can be based on the comparison between dynamical systems. Since the input feature space is a Riemannian manifold, we extend LDSs to kernel NLDSs and use kernel NLDSs to model interaction trajectories. According to the NLDSs, the similarity comparison between interaction trajectories is defined by kernels. As for kernels selection, a few strategies were proposed based on the particular symmetrical properties of the decision functions. Follow these strategies, a pairwise kernel to support pairwise classification is carefully designed.

As for experiments, three datasets have been used here. They are UT-dataset, TVHI dataset,

and the BIT dataset. The classification accuracy as well as retrieval precision comparisons between the approaches of recent publications and our method are performed and the results are shown in several tables. According to these results, the performance of our model for binary human interaction is very promising.

Compared with other approaches, our framework also take advantage of using pairwise symmetric and balanced kernels. Therefore, the training time is significant reduced because we don't need to use a symmetric training dataset which has double the size of a regular one. Additionally, our model is easy for online implementation and for extending the video analysis to multiple pairwise interactions for the purpose of analyzing person-group and group actions, and group-group interactions.

As we mentioned before, the proximity in this thesis is limited to the distance. We drop some other cues such as gaze direction or body part information. However, these information are very helpful in some datasets. For example, it is impossible to shake hands if two persons are back to back. Since there are lots of drawback in current available datasets such as no camera calibration, camera motion, difficult viewpoint, and etc, it is very difficult to estimate some information such as ground truth information, gaze direction or body part infomation, and etc. Therefore, we plan to include more cues in the future and adjust our input feature space once a new dataset is available.

References

[1] M. S. Ryoo and J. K. Aggarwal, "Recognition of high-level group activities based on activities of individual members," in *Proceedings of the 2008 IEEE Workshop on Motion and video Computing*, Washington, DC, USA, 2008, WMVC '08, pp. 1–8, IEEE Computer Society.

[2] Tian Lan, Yang Wang, Weilong Yang, and Greg Mori, "Beyond actions: Discriminative models for contextual group activities," in *Advances in Neural Information Processing Systems (NIPS)*, 2010.

[3] Ruonan Li, Rama Chellappa, and Shaohua Kevin Zhou, "Recognizing interactive group activities using temporal interaction matrices and their riemannian statistics," *Int. J. Comput. Vision*, vol. 101, no. 2, pp. 305–328, Jan. 2013.

[4] Michael S. Ryoo and Jake K. Aggarwal, "Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities," in *ICCV'09*, 2009, pp. 1593–1600.

[5] Alonso Patron-Perez, Marcin Marszalek, Ian Reid, and Andrew Zisserman, "Structured learning of human interactions in tv shows," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 12, pp. 2441–2453, Dec. 2012.

[6] Yu Kong, Yunde Jia, and Yun Fu, "Learning human interaction by interactive phrases," in *Proceedings of the 12th European conference on Computer Vision - Volume Part I*, Berlin, Heidelberg, 2012, ECCV'12, pp. 300–313, Springer-Verlag.

[7] William Brendel and Sinisa Todorovic, "Learning spatiotemporal graphs of human activities," *Computer Vision, IEEE International Conference on*, vol. 0, pp. 778–785, 2011.

[8] U. Gaur, Y. Zhu, B. Song, and A. Roy-Chowdhury, "A "string of feature graphs" model for recognition of complex activities in natural videos," in *Proceedings of the 2011 International Conference on Computer Vision*, Washington, DC, USA, 2011, ICCV '11, pp. 2595–2602, IEEE Computer Society.

[9] Gang Yu, Junsong Yuan, and Zicheng Liu, "Propagative hough voting for human activity recognition," in *Proceedings of the 12th European conference on Computer Vision - Volume Part III*, Berlin, Heidelberg, 2012, ECCV'12, pp. 693–706, Springer-Verlag.

[10] G. Johansson, "Visual perception of biological motion and a model for its analysis," *Perception, Psychophysics*, vol. 14(2), pp. 201–211, 1973.

[11] J. A. Webb and J. K. Aggarwal, "Structure from motion of rigid and jointed objects," *Artificial Intelligence*, vol. 19, pp. 107–130, 1982.

[12] J. K. Aggarwal and M. S. Ryoo, "Human activity analysis: A review," *ACM Computing Surveys (CSUR)*, vol. 43(3), 2011.

[13] Neeti A Ogale, "A survey of techniques for human detection from video," *Survey, University of Maryland*, 2006.

[14] A. Fathi and G. Mori, "Human pose estimation using motion exemplars," in *Proc. 11th Int. Conf. Computer Vision*, 2007.

[15] Greg Mori, Serge Belongie, Jitendra Malik, and Senior Member, "Efficient shape matching using shape contexts," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, pp. 1832–1837, 2005.

[16] Gregory Shakhnarovich, Paul Viola, and Trevor Darrell, "Fast pose estimation with parameter sensitive hashing," in *In ICCV*, 2003, pp. 750–757.

[17] Lubomir Bourdev and Jitendra Malik, "Poselets: Body part detectors trained using 3d human pose annotations," in *International Conference on Computer Vision (ICCV)*, 2009.

[18] Yang Wang and Greg Mori, "A discriminative latent model of image region and object tag correspondence," in *Advances in Neural Information Processing Systems (NIPS)*, 2010.

[19] Pedro F. Felzenszwalb and Daniel P. Huttenlocher, "Efficient matching of pictorial structures," in *Proc. IEEE Computer Vision and Pattern Recognition Conf.*, 2000, pp. 66–73.

[20] Deva Ramanan, "Learning to parse images of articulated bodies," in *In NIPS 2007*. 2006, NIPS.

[21] V. Ferrari, M. Marin-Jimenez, and A. Zisserman, "Progressive search space reduction for human pose estimation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[22] M.S. Ryoo, The University of Texas at Austin. Electrical, and Computer Engineering, *Semantic Representation and Recognition of Human Activities*, The University of Texas at Austin, 2008.

[23] Kris M. Kitani, Yoichi Sato, and Akihiro Sugimoto, "Recovering the basic structure of human activities from a video-based symbol string," in *Proceedings of the IEEE Workshop on Motion and Video Computing*, Washington, DC, USA, 2007, WMVC '07, pp. 9–, IEEE Computer Society.

[24] Huiqing Liu, Jinyan Li, and Limsoon Wong, "A comparative study on feature selection and classification methods using gene expression profiles and proteomic patterns," *Genome Informatics*, vol. 13, pp. 51–60, 2002.

[25] Saad M. Khan and Mubarak Shah, "Detecting group activities using rigidity of formation," in *Proceedings of the 13th annual ACM international conference on Multimedia*, New York, NY, USA, 2005, MULTIMEDIA '05, pp. 403–406, ACM.

[26] Objects Namrata Vaswani, Namrata Vaswani, and Amit Roy Chowdhury, "Activity recognition using the dynamics of the configuration of interacting," in *in IEEE Conference on Computer Vision and Pattern Recognition (CVPR*, 2003, pp. 633–640.

[27] Frédéric Cupillard, François Brémond, and Monique Thonnat, "Group behavior recognition with multiple cameras," in *Proceedings of the Sixth IEEE Workshop on Applications of Computer Vision*, Washington, DC, USA, 2002, WACV '02, pp. 177–, IEEE Computer Society.

[28] Shaogang Gong and Tao Xiang, "Recognition of group activities using dynamic probabilistic networks," in *In ICCV*, 2003, pp. 742–749.

[29] Fengjun Lv, Jinman Kang, Ram Nevatia, Isaac Cohen, and Gerard Medioni, "Automatic tracking and labeling of human activities in a video sequence," 2004.

[30] Sangho Park and J. K. Aggarwal, "Semantic-level understanding of human actions and interactions using event hierarchy," in *Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04) Volume 1 - Volume 01*, Washington, DC, USA, 2004, CVPRW '04, pp. 12–, IEEE Computer Society.

[31] Peng Dai, Huijun Di, Ligeng Dong, Linmi Tao, and Guangyou Xu, "Group interaction analysis in dynamic context," *Trans. Sys. Man Cyber. Part B*, vol. 39, no. 1, pp. 34–42, Feb. 2009.

[32] Tian Lan, Leonid Sigal, and Greg Mori, "Social roles in hierarchical models for human activity recognition," in *Computer Vision and Pattern Recognition (CVPR)*, 2012.

[33] Louis-Philippe Morency, Iwan de Kok, and Jonathan Gratch, "Context-based recognition during human interactions: Automatic feature selection and encoding dictionary," in *10th International Conference on Multimodal Interfaces (ICMI 2008)*, 2008.

[34] Biao Jin, Wenlong Hu, and Hongqi Wang, "Human interaction recognition based on transformation of spatial semantics," *IEEE Signal Process. Lett.*, vol. 19, no. 3, pp. 139–142, 2012.

[35] Wongun Choi, Khuram Shahid, and Silvio Savarese, "Learning context for collective activity recognition," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2011.

[36] Nuria M. Oliver, Barbara Rosario, and Alex P. Pentland, "A bayesian computer vision system for modeling human interactions," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, vol. 22, no. 8, pp. 831–843, 2000.

[37] Sangho Park and J. K. Aggarwal, "Recognition of two-person interactions using a hierarchical bayesian network," in *First ACM SIGMM international workshop on Video surveillance*, New York, NY, USA, 2003, IWVS '03, pp. 65–76, ACM.

[38] Harika Bharthavarapu Sajid Sharlemin Gianfranco Doretto Saeid Motiian, Ke Feng, "Pairwise kernels for human interaction recognition," *ISVC*, 2013.

[39] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele, "People-tracking-by-detection and people-detection-by-tracking," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08)*, 2008.

[40] Rizwan Chaudhry, Avinash Ravich, Gregory Hager, and Ren Vidal, "Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions," in *in In IEEE Conference on Computer Vision and Pattern Recognition (CVPR*, 2009.

[41] Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc J. Van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *ICCV'09*, 2009, pp. 261–268.

[42] N. Krahnstoever, Ming-Ching Chang, and Weina Ge, "Gaze and body pose estimation from a distance," in *Proceedings of the 2011 8th IEEE International Conference on Advanced Video and Signal Based Surveillance*, Washington, DC, USA, 2011, AVSS '11, pp. 11–16, IEEE Computer Society.

[43] Peter Van Overschee and Bart De Moor, "N4sid: Subspace algorithms for the identification of combined deterministic-stochastic systems," 1994.

[44] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto, "Dynamic textures," vol. 51, no. 2, pp. 91–109, 2003.

[45] Gene H. Golub and Charles F. Van Loan, *Matrix computations (3rd ed.)*, Johns Hopkins University Press, Baltimore, MD, USA, 1996.

[46] Antoni B. Chan and Nuno Vasconcelos, "Classifying video with kernel dynamic textures," *2012 IEEE Conference on Computer Vision and Pattern Recognition*, vol. 0, pp. 1–6, 2007.

[47] Bernhard Scholkopf and Alexander J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, MIT Press, Cambridge, MA, USA, 2001.

[48] Sajid Siddiqi, Byron Boots, and Geoffrey J. Gordon, "A constraint generation approach to learning stable linear dynamical systems," in *In Advances in Neural Information Processing Systems (NIPS)*, Vancouver, Canada, December 2007.

[49] Sung-Hyuk Cha and Sargur N. Srihari, "On measuring the distance between histograms.," *Pattern Recognition*, vol. 35, no. 6, pp. 1355–1370, 2002.

[50] Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification (2nd Edition)*, Wiley-Interscience, 2000.

[51] S. V. N. Vishwanathan, Ren Vidal, and Alexander J. Smola, "Binet-cauchy kernels on dynamical systems and its application to the analysis of dynamic scenes," *International Journal of Computer Vision*, vol. 73, pp. 2007, 2005.

[52] Fischer A. Luig K. Thies T Brunner, C., "Pairwise support vector machines and their application to large scale problems.," *JMLR*, vol. 13, pp. 2279–2292, 2012.

[53] Asa Ben-Hur and William Stafford Noble, "Kernel methods for predicting protein–protein interactions," *Bioinformatics*, vol. 21, no. 1, pp. 38–46, Jan. 2005.

[54] Carl Vondrick, Deva Ramanan, and Donald Patterson, "Efficiently scaling up video annotation with crowdsourced marketplaces," in *Proceedings of the 11th European conference on Computer vision: Part IV*, Berlin, Heidelberg, 2010, ECCV'10, pp. 610–623, Springer-Verlag.

[55] Chih-Chung Chang and Chih-Jen Lin, "Libsvm: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 27:1–27:27, May 2011.

[56] Clarke C.L.A. Cormack G.V. Buettcher, S., *Information Retrieval*, MIT Press, 2010.

[57] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li, "Learning to rank: from pairwise approach to listwise approach," in *Proceedings of the 24th international conference on Machine learning*, New York, NY, USA, 2007, ICML '07, pp. 129–136, ACM.