2014

Human Interaction Recognition with Audio and Visual Cues

Ranya Almohsen
West Virginia University

Human Interaction Recognition with Audio and Visual Cues

by

Ranya Almohsen

Thesis submitted to the
Statler 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

Hany H. Ammar, Ph.D.
Natalia A. Schmid, Ph.D.
Gianfranco Doretto, Ph.D., Chair

Lane Department of Computer Science and Electrical Engineering

Morgantown, West Virginia
2014

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

Abstract

Human Interaction Recognition with Audio and Visual Cues

by

Ranya Almohsen
Master of Science in Computer Science

West Virginia University

Gianfranco Doretto, Ph.D., Chair

The automated recognition of human activities from video is a fundamental problem with applications in several areas, ranging from video surveillance, and robotics, to smart healthcare, and multimedia indexing and retrieval, just to mention a few. However, the pervasive diffusion of cameras capable of recording audio also makes available to those applications a complementary modality. Despite the sizable progress made in the area of modeling and recognizing group activities, and actions performed by people in isolation from video, the availability of audio cues has rarely being leveraged. This is even more so in the area of modeling and recognizing binary interactions between humans, where also the use of video has been limited.

This thesis introduces a modeling framework for binary human interactions based on audio and visual cues. The main idea is to describe an interaction with a spatio-temporal trajectory modeling the visual motion cues, and a temporal trajectory modeling the audio cues. This poses the problem of how to fuse temporal trajectories from multiple modalities for the purpose of recognition. We propose a solution whereby trajectories are modeled as the output of kernel state space models. Then, we developed kernel-based methods for the audio-visual fusion that act at the feature level, as well as at the kernel level, by exploiting multiple kernel learning techniques. The approaches have been extensively tested and evaluated with a dataset made of videos obtained from TV shows and Hollywood movies, containing five different interactions. The results show the promise of this approach by producing a significant improvement of the recognition rate when audio cues are exploited, clearly setting the state-of-the-art in this particular application.

Acknowledgements

The success of any project depends largely on the advices and encouragement of many others. I take this opportunity to express my gratitude to the people who have been instrumental in the successful completion of this project. First, I would like to show my greatest appreciation to Professor Gianfranco Doretto. Words cannot express how grateful I am to him. I feel motivated and encouraged every time I attend his meeting.

I also would like to thank, Professor Hany Hammar and Professor Natalia A. Schmid being on my thesis committe and for their kindness and support.

Furthermore, I would like to thank Professor Donald Adjeroh for his precious time and encouragement during my Master.

In addition, I have to thank my lab mates, especially Saeid, who helped me throughout the process of the thesis and gave me his time and suggestions.

Special thanks go to my family, my mom and my dad for their support your: prayer for me was what sustained me thus far. My sisters, you are always in my heart.

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

Human activity recognition from video is one of the most active research areas in computer vision. During the last decade many papers have been published where a single person performs an action (e.g., walking, hand waving, eating, etc.). Those approaches were tested on artificially generated datasets. Other approaches focussed on group activity modeling and recognition.

In this work we develop an approach to recognize binary human interactions, which are a human activities that involve two persons (e.g., shaking hands, hugging, , etc.). The aim is to interpret human-to-human interactions that are captured in realistic videos, and only in the last few years, more realistic interaction datasets [3, 7, 8] have become available. Such recognition technology has been applied in many industry areas such as: security, surveillance, games, robotics, etc. We test our approach on a datasets that was obtained from TV shows and Hollywood movies. The data were divided into five interaction classes: Handshaking, High-Five, Huging, Kissing and Negative. Current approaches for interpreting such kind of interactions only use video information and discard the information encoded in the audio or, they use only audio features, and do not consider video data. Our approach is to combine audio and video to improve the recognition accuracy compared with other approaches.

In human activity recognition, the study of single person activities reveals each persons 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. In order to quickly and accurately recognize binary interactions, it is necessary to establish an efficient modeling framework.

This thesis aims at developing such a framework, leading to an approach that will achieve significant accuracy when compared with others, 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. This allows to analyze the spatio-temporal volume around each person and to extract relevant proximity cues and motion features. At the same time, the tracking information of a pair of individuals enables the extraction of audio cues, which could be coupled together with the video cues to form interaction trajectories. To make such audio and video interaction trajectories useful, this thesis models them as the output of *kernel state space (KSS)* models, and therefore reduces the problem of recognizing human interactions to the problem of discriminating between KSS models. However, this method requires to combine temporal trajectories from multiple modalities for the purpose of recognition. To this end we developed kernel-based methods for the audio-visual fusion that act at the feature level, as well as at the kernel level, by exploiting multiple kernel learning techniques.

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 the importance of audio cues are also discussed in this chapter. Chapter 3, presents a framework and principles for modeling binary interactions. Chapter 4 focuses on the kernel methods that have been designed for combining the audio and video domains, whereas Chapter 5 describes the dataset and experimental results. This chapter shows the classification accuracy of the proposed kernel methods, validating the 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. The thesis concludes in Chapter 6.

Chapter 2

Human Activity Analysis Review

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 [9]. 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 [10]. 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 perspective of video understanding is taken into account, it can be separated into four levels [1]: Object-level, Tracking-level, Pose-level, and Activity-level. From the complexity perspective, action recognition can be divided into single person action recognition, human to human interaction (also called as binary interaction) recognition, and group activity recognition. 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 for these

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 [5]

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 is present at a certain time and place. So, all people in the given video should be recognized and automatically marked, this is called people detection. The algorithms for such detection are the same used for the detection of other kinds of objects. These algorithms were classified as “based on background subtraction” and “based on direct detection” [5]. 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 in recent publications, respectively.

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 [5]

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

Besides the trajectories, human pose recognition is also an important aspect for video understanding. 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 [11]: One is matching templates which are called as exemplar-based approaches [12, 13, 14, 15]. Another one consists of fitting a human body model[16, 17, 18]. 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 into 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.

2.2 Approach-based taxonomy of human activity recognition

Single layer approaches and hierarchical approaches 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 techniques such as linear support vector machines (SVM) or hidden Markov models (HMM) to determine the classification of these unknown image sequences. In recent years, various representation types and matching algorithms have been developed under single layered approaches. Most of them adopt a sliding windows tech-

nique 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 describe later, are a 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 a data in 3D, XYT where space is the 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 framework is similar to the template matching framework which we talked about in the previous section. Another kind of single layer approaches, called 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 features extracted from each image frame describe a human statuses. 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 between the sequence and the activity class representation. 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 two other space time representations based on trajectories approaches and space time features. In trajectory approaches, an activity can be represented as trajectories in 3D space. As mentioned in the 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 enough detail analysis and results in many cases, but body parts analysis is always difficult for this kind of approaches. Instead of pure volume or pure trajectory, a set of features extracted from the volume or the trajectory is also used to represent human activity. In this kind of approaches,

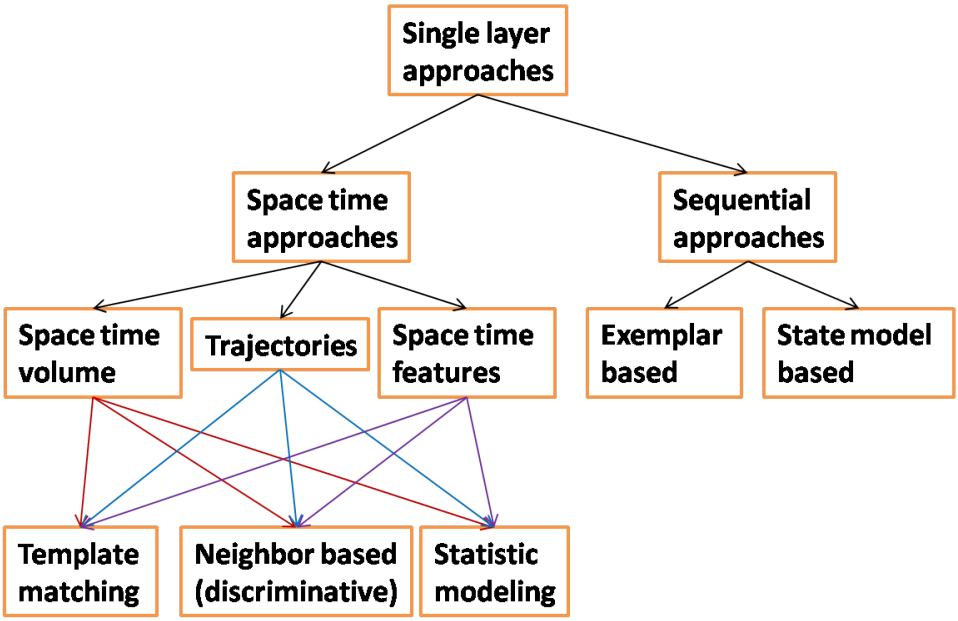

Figure 2.1: Single-layered approaches and the lists of selected publications corresponding to each category [1].

volumes or trajectories are treated as objects where common features can be extracted from them. This kind of approaches is more reliable even under noise and illumination changes. However, the computational 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 in three types: template matching, neighbor-based (discriminative), and statistical modeling. In template matching approaches, the representative models for all activities are established through 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. They can be exemplar based and state model based. A tree structure taxonomy's figure of single layer approaches is shown in Fig 2.1[1].

Another kind of approaches are the hierarchical approaches. They aim at recognizing high-

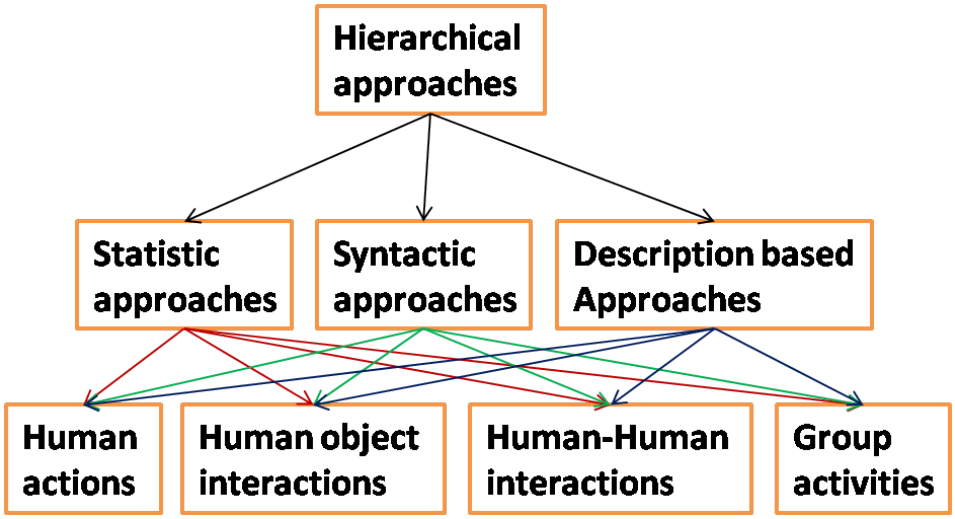

Figure 2.2: Hierachical approaches and the lists of selected publications corresponding to each category [1].

level human activities from the recognition results of other simpler activities. Intuitively, any complex 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 a higher level event derived from these known sub-events. The idea of hierarchical approaches greatly improves 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 tractable 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 of 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 numerous publications. This kind of algorithms is robust enough for activity recognition even in the case of noisy inputs. However, they 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 [19]. The same, as the case of hierarchical statistical approaches, where atomic activities are recognized by low level features. These atomic activities are then parsed to symbols through 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 that they need the recognition of the concurrent activities which is composed of concurrent sub-events. Besides that, another limitation comes from the synthetic approaches assumption. All observations are assumed to be parsed by production rules. This assumption is problematic when an unknown observation interferes with the recognition. To overcome such limitation, some algorithms are developed for automatically learning grammar rules from observations [20].

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 out 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

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

action category. 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 the previous section, some useful tools for activity recognition will be introduced in this section.

2.3.1 Detectors and descriptors 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 ST-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

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

performs much better than a traditional cuboid detector especially in noisy environments. However, the initial foreground area segmentation increases the cost of such detector. A volumetric feature detector is a detector based on Viola and Jones rectanglar features. It defines an integral video and is calculated on the x and y optical flow channels. This detector has dense features at many locations and scales resulting 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 the automatic scale selection. But examples suggest that high entropy ST-regions are rare.

Once features have been detected, extracting these features to get information from an image or video will be the next step. However, the input data is often too large to be processed. To handle redundant data, we need to transform them into a reduced representation. We call a 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 to be a good choice.

Besides a detector and a 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 vote 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 common classifiers are Support Vector Machines (SVMs). SVMs are a kind of a blend of linear modeling and instance-based learning [21]. They separate the dataset into training samples and testing samples. A linear discriminant function which is used to distinguish each class will be learned from training samples and then applied to test samples. In case there is no linear separation from training samples, SVM kernels will project the training samples onto a higher-dimensional space. Then the classifier can be learned in this high-dimensional space. K-mean is also an important classification tool. This classifier calculates the means of initial classes which are evenly distributed over the whole data space. By using a minimum distance, K-mean iteratively clusters features into the nearest classes. In each iteration, pixels/features in data space are reclassified based on previous means and then the class means are recalculated. This process continues until the number of pixels/features 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, trajectories of the group can be extracted from the video and can be compared with templates for activity analysis [22]. 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 cues for recognition[23]. In another kind of group activity, individual actions are different and each member has his/her own role. Early researches focus on the recognition of group activity by analysis of the members with non-uniform behaviors in a single group [24, 25, 26]. For example, a teacher is giving a presentation while all other students are listening in a classroom. In recent years, more challenging group activities have been analyzed. In some activities, each person has a different role. For such kind of group activities, the activity of each member in the scene has to be recognized and their structures should be analyzed. Therefore, most approaches

for the recognition of such group activity are hierarchical because there model at least two-levels of activities: group activity and each member activity [27, 28, 29]. The most popular approaches are statistical hierarchical approaches which have been discussed in the previous section. In recent years, some methodologies have been developed to handle both kinds of group activity and achieve promising results [30, 31, 32, 33, 34].

2.3.2 Binary human-human interactions

Because of the lack 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 [35]. They obtain the trajectories of both persons and compute the MLP to classify an interaction. Around 2004, J.K. Aggarwal's research group developed a hierarchical method for binary interaction recognition [36, 27]. 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 an interaction in different categories. With a new realistic dataset, this research group developed a video structure comparison method in later years [3]. This well-known new dataset is called as UT-interaction dataset. So far, it is still the most popular dataset for binary interaction studies. In their work, they extracted histogram based spatio-temporal local features from videos. After that, they create a match kernel which is a Mercer's kernel and use this match kernel to measure the similarity of feature structures from different videos. Then, they localize the detected atomic 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 can be recognized. Compared to previous works, the approach proposed in their work greatly improve the recognition accuracy for realistic binary interactions.

With more realistic datasets made available in recent years, diverse methodologies were developed. One typical volumetric-based approach is proposed by Brendel et al. in 2011 [37]. They extracted pixel intensity and motion properties at multiple scales and segment them to obtain homogeneous sub-volumes, called tubes. These tubes are organized based on their relationships: Hierarchical, Temporal, and Spatial. To simplify, they constructed a spatial-temporal graph by us-

ing nodes to represent tubes and weighted direct edges to represent these relationships. Based on these knowledge, they learned weighted least squares graph models 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 an arbitrary permutation. According to their results, the performance of this approach on the UT-interaction dataset is better than that of [3].

In the same year, Guar et al. proposed another model, the string of feature graphs model [38]. 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 is 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 have interactions between multiple objects. The experiments in their publication indicate that they achieved results comparable with [3].

In 2012, Patron-Perez et al. developed a new approach to recognize binary interactions in video from their new TVShow dataset [7]. 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 global cues. Then, they use structure SVM for learning and inference of interaction classes. Besides their new dataset, they also tested their model on the UT-interaction dataset. The classification accuracy is even better than that of Brendel's work.

With a new BIT interaction dataset, another approach was proposed by [8]. They used high-level descriptors, which are 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 a latent SVM. They tested their model on both the BIT dataset and UT dataset and got encouraging results.

Besides the approaches above, one interested approach, propagative Hough voting approach,

was proposed by Yu et al. in 2012 [39]. In their work, they use propagative Hough voting to analyze binary interactions. 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 dimensional 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 the UT dataset and TVShow dataset proves that it is an excellent methodology for binary interaction recognition.

2.4 Exploiting Audio Cues

An important step to get more advanced results in any classification problem, could be made by developing more powerful features or understanding the feature space, rather than building new classification scheme. The problem of human activity recognition has been addressed by several authors, most of them are using only one modality, which is given by the visual features, and they discard the information that is encoded by the audio. Audio is an important cue that can be exploited to improve the performance of human activity recognition.

When comparing two scenes, for example one belonging to the hugging class, and the other one belonging to the kissing class, by using only video features those two cases could be ambiguous for a computer to decide which class the scene belongs to. However, if we consider the audio signal, we notice that the hug case has a very different audio pattern than the kiss case [6].

This motivates the use of audio features. This is done by identifying the components of the audio signal that are good for identifying the linguistic content and discarding all the other stuff which carries information like background noise, etc. During the last decade, several authors have proposed algorithms to classify incoming audio data based on different algorithms. Most of these proposed systems that combine two processing stages. The first stage analyzes the incoming waveform and extracts certain parameters (features) from it. The feature extraction process usually involves a large information reduction. The second stage performs a classification based on the extracted features. A variety of signal features have been proposed for general audio classification. The most succesful one is the Mel-frequency cepstral coefficients (MFCCs). Prior to the introduction of the MFCCs, Linear Prediction Coefficients (LPCs) and Linear Prediction Cepstral

Coefficients (LPCCs) were the main feature types for speech recognition problems. The features consist of two sets: the first feature set is the low-level signal features, which include parameters such as: the zero-crossing rate, the signal bandwidth, the spectral centroid, and signal energy. The second feature set consists of Mel-frequency cepstral coefficients (MFCC). This parametric description of the spectral envelope has the advantage of being level-independent and of yielding low mutual correlations between different features. Classification based on a set of features that are uncorrelated is typically easier than that based on features with correlations. Both low-level signal properties and MFCCs have been used for general audio classification schemes of varying complexity. The simplest audio classication tasks involve the discrimination between music and speech. Typical classication results of up to 95% accuracy have been reported .When comparing the performance of the low-level signal features and MFCC features, MFCC seems to be more powerful [40].

In this thesis we will develop a framework for measuring audio and visula cues for the purpose of detecting interactions captured in video sequences. This particular area of research seams to be parctically unexplored, and our approach sets the state-of-the-art for the classification accuracy on the TVShow dataset.

Chapter 3

Human Interaction Representation

Recognition of binary interaction is one of the important areas for the automatic understanding of human activities by a computer. However, the research done in this area is much less than in other areas of human activity recognition because of the lack of realistic datasets. To improve the recognition accuracy for binary interaction, it is necessary to establish a modeling framework. In this chapter, we explain how to construct this framework and its principles. Compared with other approaches, this new framework boosts both recognition performance and efficiency for binary interaction recognition [41].

In this chapter we will describe the visual and the audio features used to represent human interactions. The temporal evolution of such features will produce interaction trajectories. Those in turn will be modeled as the output of jernel state space models, which can be compared through the use of a kernellized version of so-called Binet-Cauchy kernels. The introduction of this representation and tools is necessary to set the state for developing a kernel-based method for combining audio and visual features for human interaction recognition, as it will be explained in the next chapter.

3.1 Visual Features

Given a video, we convert it into an image sequences $\{I_t\}_{t=1}^{T}$, where t represents the frame number and T is the length of the sequence. For binary interactions, there should be two or more persons (other people will be considered as perturbation) in the image sequences. We assume the region of each person at every frame to be given through the use of a people tracker [42] This is a

Figure 3.1: An example of bounding boxes

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 inside the bounding box area instead of the whole frame region.

3.1.1 Histogram of Oriented Optical Flow

To effectively represent a binary interaction, we extract two kinds of features from the video. The first one is the *histogram of oriented optical flow* (HOOF) [2], $\mathbf{h}_{i,t}$. Here i means the i-th person in the video. Optical flow, as one of methods to detect human motion, is defined as the apparent visual motion in the scene. The second row of Figure 3.2 shows an example of an optical flow image. However, optical flow computations are sensitive to variations of scale, background noise, and the direction of movement. To overcome these problems, HOOF is based on the distribution of optical flow, as it was proposed by Chaudhry et al. in 2009. They binned the 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

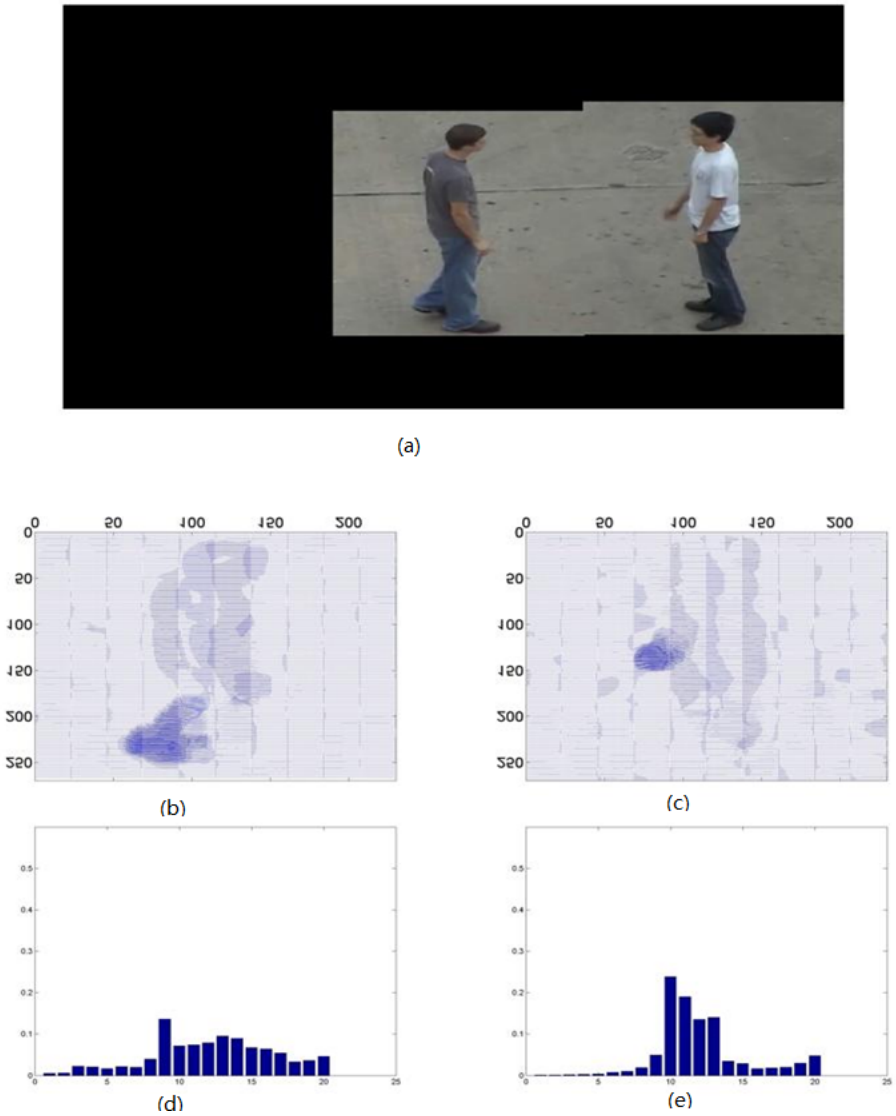

Figure 3.2: An example of HOOF descriptor. a) Binary interaction image cut from video. b) Optical flow of left person. c) Optical flow of right person. d) histogram bins obtained from b). e) histogram bins obtained from c).

obtained from the optical flow images, and Figure 3.3 shows how the histogram was formed with this method. From Figure 3.3, HOOF is symmetric in the orientation of the optical flow, which indicates that it is independent of the direction of motion. Although HOOF features can not be used to represent the relative direction of motion between pair persons, it represents each 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

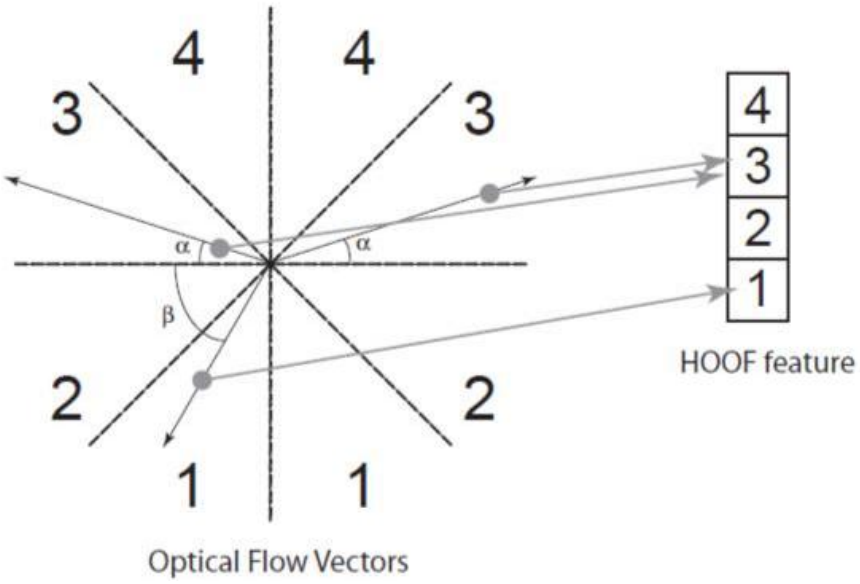

Figure 3.3: Histogram formation with 4 bins [2].

two persons will be represented by another feature.

3.1.2 Motion histograms

Another kind of features we used in this framework is called *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 is binned inside the bounding box of person to obtain the motion histogram of person i at frame t, $\mathbf{m}_{i,t}$. Like the HOOF, the MH features are also scale invariant, robust to noise, and independent of motion direction. Figure 3.4 shows a couple of examples of motion images with the corresponding MH features. Here, the vertical axis is the normalized histogram and the horizontal axis is the number of bins.

After extracting the HOOF features and the MH features, we use them to represent the person in the scene. The i-th person, can be represented by the sequence of HOOF and MH features

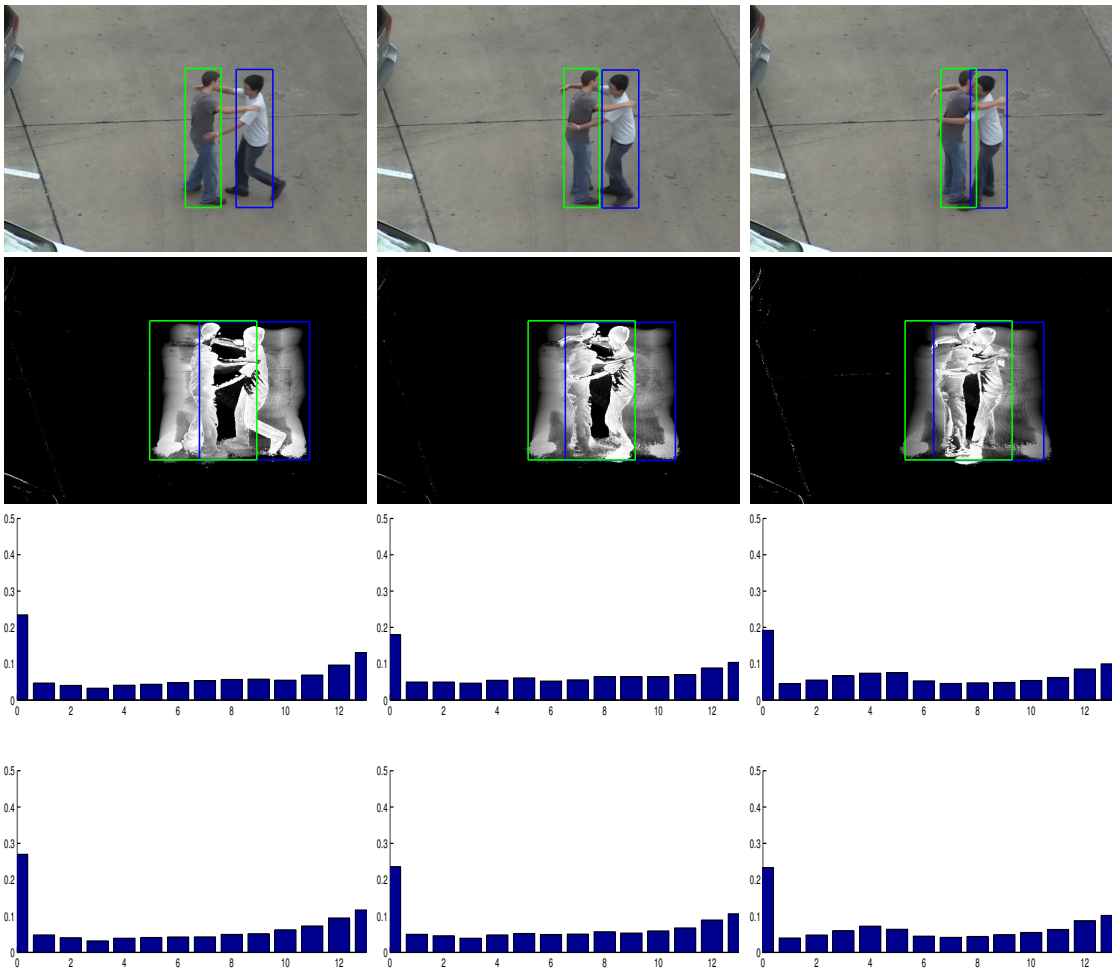

Figure 3.4: Motion images and MH feature trajectories [3]. First row: Binary interaction images obtained from video; Second row: Motion images; Third row: Motion histogram bins of left person; Fourth row: Motion histogram bins of right person.

$\mathbf{h}_t \doteq \{\mathbf{h}_{i,t}\}_{t=1}^{T}$, 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 proxemics interaction between persons also provides discriminative information (e.g., person i cannot shake hands with person j if they are far enough), and needs to be considered in this representation. Here, the spatial relationship between a pair of persons is considered. Generally, the spatial relationship could be obtained by analyzing of the Euclidean distance between the position $\mathbf{p}_{i,t}$ of person i, and the position $\mathbf{p}_{j,t}$ of person j [43].

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

The position and velocity of each person in the scene will be easily obtained if the camera calibration is known and people tracking is performed on the ground-plane. However, if the camera is not calibrated or the calibration is not with respect to the ground plane, we have to characterize proximity by approximating the distance in each frame with the distance between bounding boxes, and performing a normalization based on the person bounding box size. Even in such case where the distance is not view invariant, the experiment results for the tested datasets still show a significant improvement of the classification accuracy when the distance is considered [41].

Relative orientation between a pair persons 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 orientation or gaze direction [44]. 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 body part orientations information from video is difficult beacuse a reliable 3D articulated body tracker is required. The use of those features is beyond the scope of this thesis.

3.2 Audio Features

Extracting discriminative features is the first step to improve the results in the classification problems. State-of-the-art features for audio classification are the Mel-Frequency Cepstral Coefficients (MFCCs), which are explained in the next section.

3.2.1 Mel-Frequency Cepstral Coefficients (MFCCs)

In audio processing, Mel-frequency cepstral coefficients (MFCCs) are coefficients extracted from the Mel-frequency cepstrum (MFC), which is a representation of the short-term power spectrum of sound, after a cosine domain transform of a log-power transformation.They were introduced by Davis and Mermelstein in the 1980's, and have been state-of-the-art ever since [45]

MFCCs are derived in several steps. Since the audio signal always varies during a time period, (the change might be in the pitch or the speed of the signal), we need to divide the audio signal frame into shorter frames, and assume that during a short frame time the audio signal doesn't change much. Therfore, we divide the signal into 20-40 ms frames (25 ms is a typical choice).

If we divide the signal into shorter frames, then there are not enough samples to get a reliable spectral estimate, and if we divide the signal into longer frames, then the signal is going to change throughout the frame. Our approach is to use a temporal "sliding window" that traverses along the audio signal and computes the MFCCs. We tested several sliding window sizes start from 20 ms to 200 ms.

For each window we calculate the Discrete Fourier Transform (DFT) of the frame as follows

$$\mathbf{S}_i(k) = \sum_{i=1}^{N} S_i(n)h(n)e^{-j2\pi kn/N}, \tag{3.2}$$

where $\mathbf{S}_i(n)$ here is the time domain signal at time i, $h(n)$ is an N sample long analysis window, and K is the length of the DFT.

The next step is to calculate the periodogram estimate of the power spectrum of each frame, which is given by

$$\mathbf{P}_i(k) = {}^{1}/_{N}|\mathbf{S}_i(k)|^2, \tag{3.3}$$

Where $\mathbf{P}_i(k)$ is the power spectrum of frame i. This is derived from the human cochlea (an organ in the ear) which vibrates at different spots depending on the frequency of the incoming sounds. The calculated periodogram estimate performs a similar job by identifying which frequencies are present in the frame.

After that, we need to apply the Mel filterbank to the power spectra, because the periodogram spectral estimate contains a lot of information not useful for audio classification. Therfore, we take a group of periodogram bins and sum them up to get an idea of how much energy exists in various frequency regions. To calculate the filterbank energies we multiply each filterbank with the power spectrum, then add up the coefficents. This is performed by our Mel filterbank: the first filter is very narrow and gives an indication of how much energy exists near 0 Hertz. As the frequencies get higher our filters get wider as we become less concerned about variations. We are only interested in roughly how much energy occurs at each spot. The Mel scale tells us exactly how to space our filterbanks and how wide to make them.

When we have the filterbank energies, we take the logarithm of them, because the logarithm allows us to use cepstral mean subtraction, which is a channel normalisation technique.

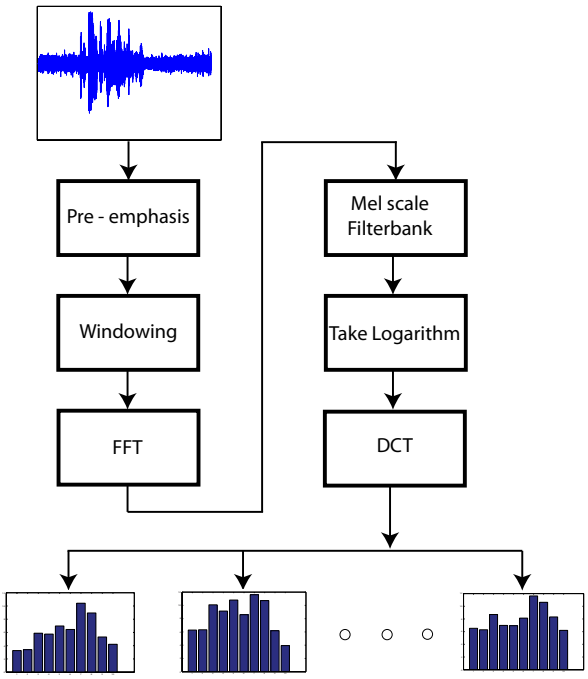

Figure 3.5: The procedure for computing MFCC features

Then, we compute the discrete cosine transform (DCT) of the list of Mel log powers. The DCT decorrelates the energies. The last step is to keep 2-13 DCT coefficients, and discard the rest. This is because the higher DCT coefficients represent fast changes in the filterbank energies, and it turns out that these actually degrade classification performance, so we get a small improvement by dropping them [46].

3.3 Audio and Video Trajectories

As anticipated in previous sections, a binary human interaction is represented by interaction trajectories. We are going to have two types of interaction trajectories. One representing visual cues (visual interaction trajectory), and one representing audio cues (audio interaction trajectory). The visual interaction trajectory consists of HOOF features (*histogram of oriented optical flow*), and MH features (*motion histogram*), that are going to be computed for each person in the bounding box over time. Therefore, visual cues are going to be represented 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}. When these quantities evolve over time we obtain a *visual interaction trajectory* which is the temporal sequence $\mathbf{y}_{ij} \doteq \{\mathbf{y}_{ij,t}\}_{t=1}^{T}$, where

$$\mathbf{y}_{ij,t} \doteq [\mathbf{h}_{i,t}^{\top}, \mathbf{m}_{i,t}^{\top}, \mathbf{h}_{j,t}^{\top}, \mathbf{m}_{j,t}^{\top}, d_{ij,t}]^{\top} . \tag{3.4}$$

The second type of interaction trajectory is the audio interaction trajectory, which consists of MFCC features. Given a video, we apply our temporal sliding window technique to the audio signal extracted from the video, and compute the MFCCs for each window from the beginning to the end of the signal. During the computation of the MFCCs we use a step size corresponding to the video frame rate, and the sliding window size is varies between 20ms to 200ms. The audio interaction trajectory is going to be a temporal sequence made of T samples, given by

$$\mathbf{a}_{ij} \doteq \{\mathbf{a}_{ij,t}\}_{t=1}^{T} \tag{3.5}$$

In summary in this thesis a human interaction is represented by a pair of interaction trajectories, indicated by (y_{ij}, a_{ij}).

3.4 Modeling Temporal Sequences

In general, an interaction trajectory $\{y_t\}$ (where here y_t indicates either visual or audio cues) 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 a binary interaction is converted to the problem of recognizing stochastic processes. Stochastic processes can be modeled as the output of dynamical systems. A dynamical 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. In this section, we will introduce linear dynamical systems and one particular non-linear estimation, which we call kernel state space models, and that we use for modeling interaction trajectories.

3.4.1 Linear Dynamical Systems (LDSs)

A linear dynamical system (LDS) is defined by the following expression:

$$\begin{cases} x_{t+1} = Ax_t + Bv_t \\ y_t = Cx_t + \mu + w_t \end{cases} \tag{3.6}$$

Here, x_t is the state of the LDS at time t. y_t is the observed output at time t. A,B,C are coefficients, where A describes the dynamics of the state evolution, B models how the state of evolution is affected by the input noise, and C transforms the state of evolution to an observation. v_t and w_t are the system noise and the observation noise. Those are independent and zero-mean, following a Gaussian distribution. μ is the mean of the past $T-1$ frames, $\{y_t\}_{t=1}^{T-1}$. These quantities are defined in the following spaces: $x_t \in \mathbb{R}^n$, $v_t \in \mathbb{R}^{n_v}$, $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times n_v}$, $y_t \in \mathbb{R}^m$, $\mu \in \mathbb{R}^m$, $C \in \mathbb{R}^{m\times n}$, $w_t \in \mathbb{R}^m$. Based on these parameters, the an LDS can be represented as $L(x_0, A, B, C, \mu, R)$ where x_0 is the initial state and R is the covariance of the observation noise. If we assume the data to be zero-mean, and if B is absorbed by the system noise distribution, model (3.6) simplifies to

$$\begin{cases} x_{t+1} = Ax_t + v_t \\ y_t = Cx_t + w_t \end{cases} \tag{3.7}$$

Now, an LDS can be represented as $L(x_0, A, C, R)$. The parameters defining the LDS can be learned from the feature trajectories of those training videos. There are several approaches to estimate these parameters. One typical method is to use the subspace identification algorithm N4SID, which is available in the Matlab toolbox [47]. 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 [48]. In this algorithm, the observations $Y_1^T \doteq [y_1, ...y_T]$ are decomposed to $U\Sigma V^T$, via singular value decomposition (SVD) [49], with $\Sigma = diag\{\sigma_1, ..., \sigma_n\}$. Therefore, the parameters are estimated as $\hat{C}(T) = U$, and $\hat{X}(T) = \Sigma V^T$. $\hat{A}$ can be determined uniquely by solving

$$\hat{A}(T) = \Sigma V^T D_1 V (V^T D_2 V)^{-1} \Sigma^{-1} \tag{3.8}$$

where $D_1 = \begin{bmatrix} 0 & 0 \\ I_{T-1} & 0 \end{bmatrix}$ and $D_2 = \begin{bmatrix} I_{T-1} & 0 \\ 0 & 0 \end{bmatrix}$. $\hat{B}$ is determined by input noise covariance Q by $\hat{B}\hat{B}^T = \hat{Q}$. A more detailed derivation and implementation of this algorithm is given in [48].

After these parameters are determined, similarity between different LDSs will be defined through kernels such as Binet-Causchy kernels, RBF kernels, string kernels, etc. Based on a specific kernel, all similarities of training data will be computed and used for testing data classification through a support vector machine (SVM) classifier.

3.4.2 Kernel State Space Models

So far, we described the LDSs approach and the algorithm for parameters estimation. However, if the space $\mathcal{S}$ where y_t is not euclidean, representing $\{y_t\}$ within LDS is suboptimal. One way to proceed in this case is suggested in [2, 4]. Instead of using PCA to learn a linear observation function in LDSs, they use kernel principle components analysis (KPCA) to learn a non-linear observation function. Therefore, we refer to such dynamical system as kernel state space model, which we now introduce.

To understand KSSs, it is necessary to introduce KPCA first. Kernel PCA is the kernelized version of standard PCA [50, 4]. As we know, the data is projected into a linear principal component in standard PCA. In KPCA, the data is projected onto a non-linear subspace and those non-linear principle components are expressed by kernel function. That means KPCA performs a non-linear feature transformation of 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 $\Phi(\cdot) : \mathcal{S} \to \mathcal{H}$, 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 [4] for a detailed description and derivation).

Based on the KPCA introduction, now we consider the extension of LDSs to KSSs. As we mentioned before, KPCA first transforms the data with the feature transformation $\Phi(\cdot)$ which is induced by the kernel function $K_(y_s, y_t) = \Phi^T(y_s)\Phi(y_t)$, and then a standard PCA is used as it is done in LDSs. So an observation sequences $\mathbf{y}_t$ can be transformed to $\Phi(\mathbf{y}_t)$. Therefore, the LDS equation is replaced by

$$\begin{cases} x_{t+1} = Ax_t + v_t \\ \Phi(y_t) = Cx_t + w_t \end{cases} \tag{3.9}$$

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 3.6: Parameter estimation algorithm for KSS models [4]

Compared with equation (3.7), $\Phi(\cdot)$ because the mapped space $\mathcal{H}$ could be an infinite dimension space, C is a linear operator instead of a matrix, where $C : \mathbb{R}^n \rightarrow \mathcal{H}$. To estimate the parameters of model (3.9), we need to identify the parameter A, the sequence x_t, and some representation for C based on the knowledge of kernel K. The parameter estimation algorithm is summarized in Figure 3.6.

3.4.3 Stability of LDSs and KSSs

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.

For a 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 [51]. 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 ensures the stability of the system [51]. Thus, the stability of the LDS model for a binary human-human interaction problem is ensured.

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

3.5 Comparing Trajectories with KSS Models

To classify human activity, we need to evaluate the similarity of interaction trajectories for both audio and visual cues. That means, a method to compute the similarity between KSSs has to be developed. In this thesis, the kernel we used for interactions comparison is the kernellized version of the Binet-Cauchy kernel, that was proposed in [2]. In particular, the Binet-Cauchy trace kernel for KSS 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

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

Where, $0 < \lambda < 1$, 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. 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 $t \in \{1, \cdots, T'\}$. Following the same procedure for LDSs, P can be computed by solving the 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 [2]

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}_{BC}, which is a kernel only on the dynamics of the KSS, and is given by the maximum singular value of P, i.e.,

$$K^{\sigma}_{KSS} = \max \sigma(P) \ . \tag{3.12}$$

For more details about the estimation of the KSS model parameters, and about the derivation of kernel (3.12) the reader is referred to [48, 4, 2].

Chapter 4

Human Interaction Recognition

In the previous chapter, we establish a framework for modeling binary interaction based on interaction trajectories. In this framework, the similarity between two videos can be compared through kernels. Therefore, the performance of such framework greatly depends on how well the kernel methods are designed. In this chapter, we will introduce appropriate kernel methods for comparing visual trajectories, audio trajectories, and for combining the discriminative inforamtion carried by audio and video cues. In particular we will propose the direct sum combination of audio and video trajectories, and a method based on multiple kernel learning.

4.1 Modeling Challenges

To model binary interactions, we have to address a few unique issues.

4.1.1 Domain definition of the audio and video trajectory

The measurements of the visual interaction trajectories $\mathbf{y}_{ij,t}$ do not live in an Euclidean space. As we mentioned in previous sections, the usual 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. Similarly, given the non-linear processing pipeline for computing MFCC features, $a_{ij,t}$ does not assume values in an euclidean space.

4.1.2 Interaction symmetry

The decision function of any classifier is expected to be symmetrical and should not be affected by any person ordering. This is relate 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.,

$$f(\mathbf{h}_i, \mathbf{m}_i, \mathbf{h}_j, \mathbf{m}_j, d_{ij}) = f(\mathbf{h}_j, \mathbf{m}_j, \mathbf{h}_i, \mathbf{m}_i, d_{ji}) \ . \tag{4.1}$$

4.1.3 Audio and Visual Trajectory Combination

The information carried by the audio trajectory $\{a_{ij,t}\}$, and by the visual trajectory $\{y_{ij,t}\}$ must be combined in a way that maximizes the human interaction classification accuracy.

4.2 Kernel for the Audio Domain

The audio interaction trajectory is a temporal sequence, and in order to compare two audio trajectories we need to compare their corresponding kernel state space models. Kernel functions can be used in many applications as a simple bridge from linearity to non-linearity. In this section, we are going to propose several kernels that will be used during our experiments.

4.2.1 Binet-Cauchy Kernels

We model audio interaction trajectories as the output of kernel state space models, and reduce the problem of recognizing human interactions to the problem of discriminating between KSSs. Once we have represented each audio as an interaction trajectory model, we need a dissimilarity

metric or distance to assess how close two given trajectories are. Several methods can be found in the literature, such as algebric and information theoretic distances. Besides distances, recently Binet-Cauchy kernels for computing similarity have been introcuded for LDSs, and then extended for KSSs. The Binet-Cauchy kernel for NLDS is given by:

$$K_{BC}(\{y_t\}_{t=1}^{\infty}, \{y'_t\}_{t=1}^{\infty}) \doteq E\left[\sum_{t=1}^{\infty} \lambda^t \Phi(y_t)^\top \Phi(y'_t)\right] = E\left[\sum_{t=1}^{\infty} \lambda^t K(y_t, y'_t)\right], \tag{4.2}$$

where $0 < \lambda < 1$, 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 K_{BC} can be computed in closed form, and it requires the computation of the infinite sum. In order to compute the infinite sum of the inner products $K(y_t, y'_t)$ which represent the kernel between audio frames of two trajectories, we tried the linear kernel, and the radial basis function (RBF) kernel.

- Leaner Kernel

  The linear kernel is the simplest kernel that can be used with a classifier such as a SVM. It is optimal if the data are linearly separable. To compare the frames between two audio frames, we extract the MFCC features; which will create the audio trajectory. Therfore, if y_t is A vector that contains the extraxted features $a_{ij,t}$, then the linear kernel is given by the inner product $< t, t' >$, plus an optional constant, i.e.

$$K(y_t, y'_t) = y_t^T y'_t \tag{4.3}$$

- RBF Kernel

  The RBF kernel is one of the most popular kernels, it iis often considered as the first choice. Unlike the linear kernel, the RBF kernel handles the cases when the relationship between class lables and attribures is nonlinear. Furthermore, the RBF kernel has less tuning parameters if compare it with others, which positively influences the complexity of model selection. The standard RBF kernel on two samples y_t and y'_t isdefined as :

$$K(y_t, y'_t) = \mathrm{e}^{-\gamma \|y_t - y'_t\|^2}, \tag{4.4}$$

  Where $\|.\|$ indicates the euclidean norm.

4.2.2 RBF Kernel with Binet-Cauchy Kernel Distance

Another strategy for using an RBF kernel is to define a distance between two trajectories, and then use it in place of the typical euclidean norm. In this method we first compute the Binet-Cauchy kernel for audio trajectories using equation kernel (4.2). Specically, $K(\{y_t\}, \{y'_t\}) = E\left[\sum_{t=1}^{\infty} \lambda^t K(y_t, y_t)\right]$. Then, we compute distance from the Binet-Cauchy kernel, given by

$$d_{BC}(\{y_t\}, \{y'_t\}) = K_{BC}(\{y_t\}, \{y_t\}) + K_{BC}(\{y'_t\}, \{y'_t\}) - 2K_{BC}(\{y_t\}, \{y'_t\}), \tag{4.5}$$

We refer to (4.5) as Binet-cauchy kernel distance.

Finally, we compute the following RBF kernel:

$$K_{RBF-BC}(\{y_t\}, \{y'_t\}) = \mathrm{e}^{-\gamma d^2_{BC}(\{y_t\}, \{y'_t\})} \tag{4.6}$$

4.2.3 RBF Kernel with Martin Distance

By treating the audio trajectories, $\{y_t\}_{t=1}^{T}$, as the output of LDSs we can compare two LDSs with algebraic distances, such as the martin's distance. The output trajectory $\{y_t\}$ of a LDS lives in the observability subspace associated with the model parameters, $M = (A; C)$. The observability subspace is the range-space of the extended observability matrix $\mathcal{O}(M) = [C^T, (CA)^T, (CA^2)^T, ..]^T \in \mathbb{R}^{\infty \times n}$. The martin's distance is based on the computation of the principal angles extened observability matrices, which are called subspace angles. More specifically, let $M_i = (A_i, C_i)$ for $i = 1, 2$ be the parameters of two LDS models for order n. let $\theta_1, \cdots \theta_{2n}$ be the subspace angles between the range spaces of their extended observability matrices $\mathcal{O}_1$ and $\mathcal{O}_2$, which are defined as

$$\mathcal{O}_i = \left[\mathbf{C}_i^T, (\mathbf{C}_i\mathbf{A}_i)^T, (\mathbf{C}_i\mathbf{A}_i^2)^T, \cdots\right], \quad i = 1, 2. \tag{4.7}$$

If the systems are stable, i.e., $\|\mathbf{A}_i\|_2 < 1$, the subspace angles θ_i can be computed as the roots of $\theta_i = \cos^{-1}(\sqrt{\lambda_i})$, where λ_i is the i-th eigenvalue of $\mathbf{P}_{11}^{-1}\mathbf{P}_{12}\mathbf{P}_{22}^{-1}\mathbf{P}_{21}$ AND P_{ij} is the solution to the Sylvester's equation

$$\mathbf{P}_{ij} = \mathbf{A}_i^T\mathbf{P}_{ij}\mathbf{A}_j + \mathbf{C}_i^T\mathbf{C}_j, \quad i, j = 1, 2. \tag{4.8}$$

One can show that the subspace angles are invariant with respet to a change of basis in the state space. Thus, as described in [52], one can define many distances based on the subspace angles. For example, the (squared) Martin and Forbenius distance between the modles M_1 and M_2 are, respectively, given by:

$$d_M^2(M_1, M_2)^2 = -\log \prod_{i=1}^{2n} \cos^2 \theta_i \tag{4.9}$$

$$d_F^2(M_1, M_2)^2 = 2 \sum_{i=1}^{2n} \sin^2 \theta_i \tag{4.10}$$

Finally, we use the martin diatance to derive the following RBF kernel

$$K_{RBF-M}(\{y_t\}, \{y'_t\}) = \mathrm{e}^{-\gamma d_M^2(M1,M2)} \tag{4.11}$$

4.3 Kernels for the Visual Domain

For the visual trajectories we define Binet-cauchy kernels and RBF kernels with Binet-cauchy distance as explained in the following sections.

4.3.1 Binet-Cauchy Kernels

As it was done for the audio trajectory, we compare the KSS models of visual trajectories with Binet-cauchy kernels like the one in equation (4.2). However, in order to address the modeling challenges outlined in section 4.1.1 and 4.1.2 we need to carefully design the kernel K, inside the K_{BC} kernel. this is explained in the following two sections.

Kernels for Histogrmas

Mercer kernels, proposed in [50], are positive definite kernels that induce an inner product in a higher dimensional space, called a Reproducing Kernel Hilbert Space(RKHS). For points lying on the non-linear manifold, the Mercer kernel is given by

$$k(h_1, h_2) = \Phi(h_1)^\top \Phi(h_2) \tag{4.12}$$

There are several mercer kernels for histograms. If we represent a histogram as its square root, we have $\sqrt{h_t} = [\sqrt{h_{t;1}}, ..., \sqrt{h_{t;N}}]$. Such histogram can be projected to a N dimensional hypersphere where the Riemannian metric between two points on the hypersphere induces the following kernel between two histograms.

$$k_S(h_1, h_2) = \sum_{i=1}^{N} \sqrt{h_{1;i} h_{2;i}} \tag{4.13}$$

This kernel (4.13) is known as the geodesic kernel and can be derived from the RBF kernel $k(h_1, h_2) = \exp(-d(h_1, h_2))$ with the Bhattacharyya distance $d_B(h_1, h_2) = -ln(BC(h_1, h_2))$, where $BC(h_1, h_2) = \sum_{i=1}^{N} \sqrt{h_{1;i} h_{2;i}}$.

The Bhattacharyya distance measures the similarity of two discrete or continuous probability distributions [2]. Another kind of distance to measure the similarity of histograms is the Minimum Difference of Pairwise Assignment [53], given by

$$d_{MDPA}(h_1, h_2) = \sum_{i=1}^{N} \left| \sum_{j=1}^{i} (h_{1;i} - h_{2;i}) \right|. \tag{4.14}$$

Another popular distance between two histograms is the χ^2 distance

$$d_{\chi^2}(h_1, h_2) = \frac{1}{2} \sum_{i=1}^{N} \frac{|h_{1;i} - h_{2;i}|^2}{h_{1;i} + h_{2;i}}. \tag{4.15}$$

All of these distances can be used in combination with aRBF kernel to compute the similarity of histograms.

Besides RBF kernels, another kind of Mercer kernel for histograms is Histogram Intersection Kernel (HIST) [54], which is defined as

$$k_{HIST}(h_1, h_2) = \sum_{i=1}^{N} min(h_{1;i}, h_{2;i}). \tag{4.16}$$

Pairwise Kernels

In our framework, the visual 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 in equation (3.10) should be definedaccordingly. There are several ways to construct a non-linear kernel. One way is to extend an 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 with a distance for the manifold $\mathcal{S}$. However, defining a distance on $\mathcal{S}$ is an open problem. An alternative approach is to use kernel construction techniques which are discussed in [50]. Since, $\mathcal{S}$ is represented by the Cartesian product of subspaces, this approach allows to 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. Since the input feature space $\mathcal{S}$ is represented by the Cartesian product of subspaces, we design K_H for the first subspace $\mathbb{H}_b \times \mathbb{H}_\tau \times \mathbb{H}_b \times \mathbb{H}_\tau$, and K_d for the second subspace $\mathbb{R}_+$. Following the method proposed in [50], K_H and K_d can then be combined by computing their tensor product kernel, which is expressed as

$$K \doteq (K_H \otimes K_d)(\mathbf{y}_{ij}, \mathbf{y}'_{ij}) = K_H((\mathbf{h}_i, \mathbf{m}_i, \mathbf{h}_j, \mathbf{m}_j), (\mathbf{h}'_i, \mathbf{m}'_i, \mathbf{h}'_j, \mathbf{m}'_j))K_d(d_{ij}, d'_{ij}) \ , \tag{4.17}$$

To lighten the notation, the time subscript t is not shown in the above equation. With this kernel, the classification of binary interactions is decided not only by the similarity of motion features but also by the similarity of proximity cues, as it is explained in the previous chapter.

Now, let's consider K_H and K_d separately. From Equation 4.17, K_d depends on the distance between person i-th and person j-th, $d_{ij} \in \mathbb{R}_+$, and we simply chose a gaussian RBF kernel, given by

$$K_d(d_{ij}, d'_{ij}) \doteq \exp(-\gamma |d_{ij} - d'_{ij}|^2) \ . \tag{4.18}$$

For kernel K_H, we note that it is a so-called pairwise kernel [55], because it is such that $K_H : (\mathcal{X}_H \times \mathcal{X}_H) \times (\mathcal{X}_H \times \mathcal{X}_H) \to \mathbb{R}$, where $\mathcal{X}_H \doteq \mathbb{H}_b \times \mathbb{H}_\tau$, and it could be used to support pairwise classification, which aims at deciding whether the examples of a pair $(a, b) \in \mathcal{X}_H \times \mathcal{X}_H$ belong to the same class or not.The requirement of being positive semidefinite implies that K_H satisfies the following symmetry property

$$K_H((a, b), (a', b')) = K_H((a', b'), (a, b)) \ , \tag{4.19}$$

for all $a, b, a', b' \in \mathcal{X}_H$. 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

$$K_H^D = (k_H \oplus k_H)(a, b, a', b') = k_H(a, a') + k_H(b, b') \ , \tag{4.20}$$

$$K_H^T = (k_H \otimes k_H)(a, b, a', b') = k_H(a, a')k_H(b, b') \ , \tag{4.21}$$

which obviously satisfy the symmetric property. We now verify whether by using the kernels defined in (4.20) and equation (4.21) it is possible to construct a decision function f for interaction trajectories, which are supposed to satisfy the symmetry property (4.1). We plan to learn decision function f with a SVM that exploit the general kerne equation (3.10). Therefore, they will assume the form

$$f(\{a_{i,t}, a_{j,t}, d_{ij,t}\}) \doteq \sum_{u,v} \alpha_{uv}\ell_{uv}K_{NLDS}(\{a_{i,t}, a_{j,t}, d_{ij,t}\}, \{a'_{u,t}, a'_{v,t}, d'_{uv,t}\}) + \beta \ , \tag{4.22}$$

where, α_{uv}, ℓ_{uv}, and β are the usual SVM parameters [50], 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 4.22 indicates that the symmetry property of 4.1 should be expressed as

$$\begin{aligned} K_{NLDS}(\{a_{i,t}, a_{j,t}, d_{ij,t}\},&\{a'_{u,t}, a'_{v,t}, d'_{uv,t}\}) = \\ &K_{NLDS}(\{a_{j,t}, a_{i,t}, d_{ji,t}\}, \{a'_{u,t}, a'_{v,t}, d'_{uv,t}\}) \ , \end{aligned} \tag{4.23}$$

for all $a_{i,t}, a_{j,t}, a'_{u,t}, a'_{v,t} \in \mathcal{X}_H$, and $d_{ij,t}, d'_{uv,t} \in \mathbb{R}_+$. In turn (4.23) induces a symmetry property on the kernel (4.17) through (3.10), which is given by

$$K((a_{i,t}, a_{j,t}, d_{ij,t}), (a'_{u,t}, a'_{v,t}, d'_{uv,t})) = K((a_{j,t}, a_{i,t}, d_{ji,t}), (a'_{u,t}, a'_{v,t}, d'_{uv,t})) \ , \tag{4.24}$$

and finally, since $d_{ij,t} = d_{ji,t}$ and $d_{uv,t} = d_{vu,t}$ (4.24) impose on K_H the following relationship

$$K_H((a_{i,t}, a_{j,t}), (a'_{u,t}, a'_{v,t})) = K_H((a_{j,t}, a_{i,t}), (a'_{u,t}, a'_{v,t})) \ , \tag{4.25}$$

to be valid for all $a_{i,t}, a_{j,t}, a'_{u,t}, a'_{v,t} \in \mathcal{X}_H$. Note tha the relationship (4.25) is different than the symmetry relationship (4.19), and kernels that satisfy (4.25) are called balanced [55]. Unfortunately, the pairwise kernels K_H^D, and K_H^T, defined in (4.20) and (4.21) symmetric but not balanced. Therfore, we propose to test two kernels that have been proved to to have good theoretical properties [55], 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

$$K_H^{DS}((a,b),(a',b')) = K_H^{SD}((a,b),(a',b')) + K_H^{ML}((a,b),(a',b')) \ , \tag{4.26}$$

$$K_H^{TL}((a,b),(a',b')) = \frac{1}{2}(k_H(a,a')k_H(b,b') + k_H(a,b')k_H(b,a')) \ , \tag{4.27}$$

where

$$K_H^{SD}((a,b),(a',b')) = \frac{1}{2}(k_H(a,a') + k_H(a,b') + k_H(b,a') + k_H(b,b')) , \quad (4.28)$$

$$K_H^{ML}((a,b),(a',b')) = \frac{1}{4}(k_H(a,a') - k_H(a,b') - k_H(b,a') + k_H(b,b'))^2 . \quad (4.29)$$

In particular, K_H^{TL} is called tensor learning pairwise kernel [56], and K_H^{DS} is called direct sum pairwise kernel [55].

Finally, wer left with the task of step designing k_H, which 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 concur 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

$$k_H((\mathbf{h}_{i,t}, \mathbf{m}_{i,t}), (\mathbf{h}'_{i,t}, \mathbf{m}'_{i,t})) = k_h(\mathbf{h}_{i,t}, \mathbf{h}'_{i,t}) k_m(\mathbf{m}_{i,t}, \mathbf{m}'_{i,t}) . \quad (4.30)$$

Both k_h and k_m are kernels for comparing histograms. There are several options in this domain, as it is outlined in section 4.3.1 and for both k_h and $k_m$we picked the geodesic kernel (4.13).

4.3.2 RBF Kernel with Binet-Cauchy Kernel Distance

Similarly to what was done for the audio trajectories, we define an RBF kernel for visual trajectories that is based on the Binet-cauchy kernel distance.

4.4 Kernels for the Audio and Video Domain

In this section we present several kernel-based strategies for combining audio and visual trajectories.

4.4.1 Direct Sum of audio and visual features

The experiments show that the direct sum of audio and visual features spaces display better results than only using video features or only audio features. Thus, we extract audio features consisting of vector with length between 7 to 25 elements. However, we obtained the best result with 9 elements,thus we extract MFCC features made of 9 elements. In addition; we extract the

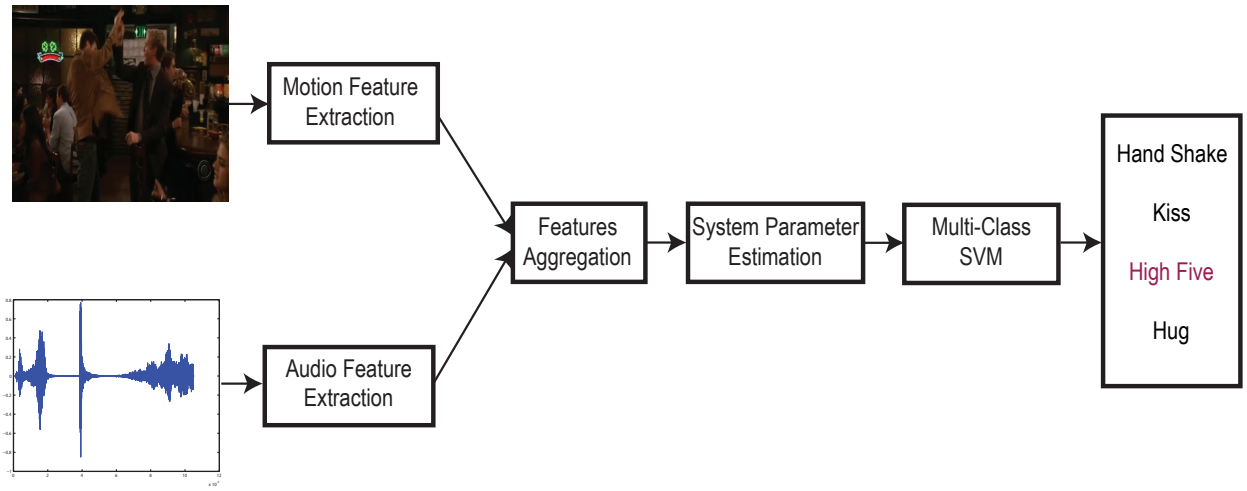

Figure 4.1: Combination of audio and video Features

video features HOOF and MH, and the distance d. The video features vector was a dimensionality between 19 to 35.

To allow the direct sum of audio and visual features, we need to ensure identical feature extraction rates. Therefore the audio features are temporally sampled so as to reach the same frame rate as the audio features. Finally the audio and visual features are concatenated to obtain a feature vector of dimensionality 28 to 44, which is used for training and testing. Figure 4.1 presents the direct sum procedure for combining audio and visual features.

4.4.2 Binet-Cauchy Kernel

As mentioned in the previous sections Binet-Cauchy kernels can be used to assess the similarity btween pairs of audio interaction trajectories and video interaction trajectories. After the direct sum of audio and video features, our trajectory will consist of $\{(y_{ij,t}, a_{ij,t})\}$. We use the BC kernel (3.6) to compare the similarity between audio and video trajectories. As internal kerneL $K(y_t, y'_t)$ we use the tensor learning pairwise kernel (4.29) to compare frames of video trajectories, for the distance d we use the RBF kernel (4.20) and for MFCC we use the RBF kernel (4.4).

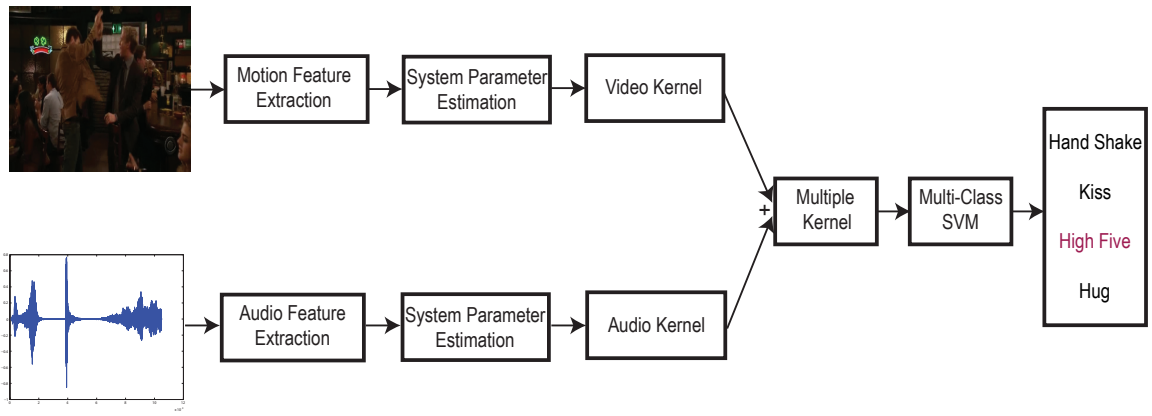

Figure 4.2: The MKL procedure for combining audio and visual

RBF Kernel with Binet-Cauchy Kernel Distance

Another kernel that we use to compare audio and video trajectory is the RBF with the Binet-kernel distance. In this method we first compute the Binet-Cauchy kernel for the audio and video trajectories using (4.2), and again we use the tensor learning pairwise kernel (4.29) to compare frames of video trajectories, for distance d we use the RBF kernel and for MFCC we also use (4.4).

Then, we create the corresponding Binet-cauchy kernel distance, and we apply the RBF kernel as it is done in (4.6).

RBF kernel with Martin Distance

As mentioned eralier, by treating the representative feature trajectories, $\{y_t\}_{t=1}^{T}$, as the output of a LDS we can compare two LDS with the martin distance. After the direct sum of the audio and video features, we estimate the corresponding LDS.

The parameters are used to compute the martin distance d_m, and we use the RBF kernel (RBF-M) inside the SVM classifier.

4.4.3 Audio Visual Multiple Kerenl Learning

Multiple kernel learning (MKL) searches for a structure to better quantify the similarities and differences between trajectories. This is done by appropriately combining multiple predefined kernels instead of using a single one. There can be two uses of MKL :

- **First:** different kernels correspond to different notions of similarity and instead of trying to find which one works best, a learning method does the picking for us, or may use a combination of them. Using a specific kernel may be a source of bias, and allowing a learner to choose among a set of kernels, should leaded to a better solution.

- **Second:** different kernels may be using inputs coming from different representations possibly from different sources or modalities. Since these are different representations, they have different measures of similarity corresponding to different kernels. In such a case, combining kernels is one possible way to combine multiple information sources. This method of combining kernels is called intermediate combination, and is in contrasts with an early combination (where features from different sources are concatenated and fed to a single learner) and late combination (where different features are fed to different classiers whose decisions are then combined by a fixed or trained combiner). Figure 4.2 shows the procedure that has been used to combine two kernels [57].

According to our experiments, the best performance when comparing audio and also visual trajectories is obtain by using RBF kernel with BC kernel distances. Thus, when combining these two kernels the final kernel is going to be $K_{total} = \eta_1 K_{audio} + \eta_2 K_{video}$, K_{total} in general is compute as the following:

$$K_\eta(X_i, Y_j) = f_\eta(\{K_m({X_i}^m, {Y_j}^m)\}_{m=1}^P|_\eta) = \sum_{m=1}^{P} \eta_m K_m({X_i}^m, {Y_j}^m) \tag{4.31}$$

where η denotes the kernel weights, which are estimated simultaneously with the other SVM classifier parameters. In general, in (4.31) f_η indicates a parametric function for the combination of kernels, but in this thesis we limit our choices to the linear version, as indicates in the right hand side of (4.31).

Chapter 5

Experimental Results

In this chapter, we present the results of the experiments conducted to compare several methods that have been used for human interaction recognition, and establish the most effective one. Our goal is to confirm that audio features can be employed to improve the classification performance.

5.1 Dataset

The dataset that has been used for our experiments is the TVShow dataset. It consists of videos that belong to five different classes: *hand-shakes*, *high-fives*, *hugs*, *kisses*, and *negative*. Each video clip is labled with a single interaction class from the possible five.There is 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. The dataset provides information about the frame intervals where the interaction happens within each video. 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 some of the original videos where not considered due to their very limited length. For the purpose of comparison, we use 120 videos. When we train samples, every time we train with 119 videos and leave one video out, then we repeat this procedure with all other videos.

To process this 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 out in Chapter 1, we assume that correct tracking information is available in our experiments. This is a common assumption in human activity analysis. Figure 3.2 and Figure 3.4 give examples of how we process these datasets. We use bounding boxes to tightly bound each person in the scene at each frame to compute the MH and the HOOF features, and we use a temporal sliding window to compute the MFCCs for each corresponding video frame. In general we use bounding boxes with a width that is three times the width of the original tight bounding box. This process is shown in the second row of Figure 3.4. Even though for the TVShow dataset we can directly exploit the included annotation information, 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 in all the expermints. Proximity cues were obtained by computing the distance between the bounding boxes. 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 in the recognition accuracy by different kernel constructions which were proposed in previous sections. Several possible choices of K_{KSS} are evaluated, and for each kind we compute the recognition accuracy. Also we compare our results with other methods that have been proposed in literature, such as the bag-of-word model [6].

The experiments include three main parts:

- Results on audio based interaction recognition
- Results on video based interaction recognition
- Results on audio-video based interaction recognition

Summary of the Bag of Word Approach				
Approach/Feature	HOG	HOF	HOG+HOF	MFCC
Audio	-	-	-	**48.5**
Video	39.5	45.0	**46.0**	-

Table 5.1: Classification accuracy, the BoW approach [6].

5.2.1 Results on Audio Based Interaction Recognition

In this section we present the experiments conducted to classify human interactions based on audio features. This includes comparison with the Bag-of-Word (BoW) approach, and show the sensitivity of accuracy with respect to number of MFCC features and temporal window size test.

Summary of the baseline Bag of Word (BoW) Approach

One method that has been used for comparison purpose is the Bag-of-Word (BoW) [6]. The general idea of the BoW model is tobuild a histogram h with k bins where each bin represents how many timeas a visual word is present in the target image. For a given video sequence the BoW model build such an audio descriptor or video descriptor, depending on the kind of dictionary that is uesd. For video experiments Spatio-Temporal Interst Points have been computed from Histograms of Oriented Gradients (HOG) and Histograms of Optical Flow (HOF). For audio experiments, the dictionary is built out if MFCC features. Table 5.1 shows a summary of the classification accuracy of the BoW aprroach.

Results on Audio Based Interaction Recognition Using our Approach

In this section we present the experiments conducted to validate the effectiveness of the audio features. Our samples have been taken from the TVShow dataset by extracting the audio wave and then computing the MFCC as audio features. Our classification scheme is based on a multi-class SVM classifier, which we have trained using the libSVM package, and for which we have tested the following kernels and model configurations:

(a) K_{KSS} as Binet-Cauchy kernel to compare two audio interaction trajectories, and for $K(y_t, {y'}_t)$ we tested the linear kernel and the RBF kernel option.

(b) RBF kernel with BC kernel distance, while $K(y_t, {y'}_t)$ has set to be either the linear kernel

Results on Audio Based Interaction Recognition		
KSS / $K(y_t, y'_t)$	Linear	RBF
Binet-Cauchy (BC)	45	44.16
RBF with BC Kernel Distace	45.83	**48.33**

Table 5.2: Audio classification accuracy based on the KSS model

Results on Audio Based Interaction Recognition	
K_{LDS}	Accuracy
RBF with Martin Distance d_m	47.5

Table 5.3: Audio classification accuracy based on the LDS model

Methods	Accuracy
Our Method	48.33
BoW	48.5

Table 5.4: Comparison between our method and the BoW model audio based interaction recognition.

or the RBF kernel. (c) RBF kernel with Martin distance d_m.

Table 5.2, Table 5.3 and Table 5.4 show the classification accuracy for the THVShow-Interaction dataset by using kernel option outlined above. We notice that the best result is obtained by using the RBF kernel with BC kernel distance.

We also plot the confusion matrices for audio obtained with the best performance kernel, (see the left of Figure 5.5).

5.2.2 Results on Video Based Interaction Recognition

Similarly to the audio case, we tested the kernel's impact on the classification accuracy based on the video cues. Figure 5.3 shows the comparison between the Binet-Cauchy (BC) kernel and the RBF kernel with BC kernel distace used for classification visual trajectories.

It is abvious that RBF kernel with BC kernel distace and tensor learning (TL) pairwise kernel gives the best result.

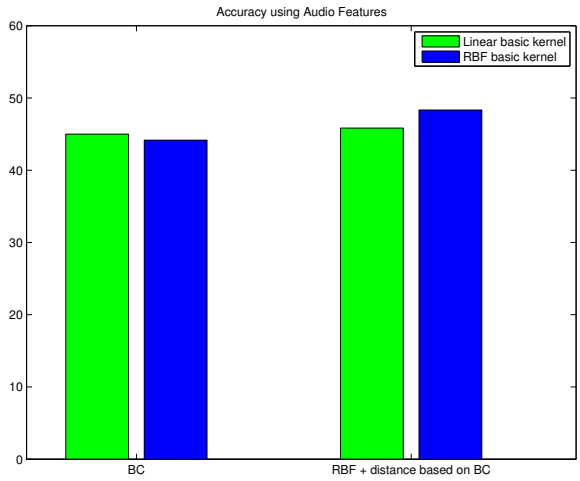

Figure 5.1: Classification accuracy summary based on audio features.

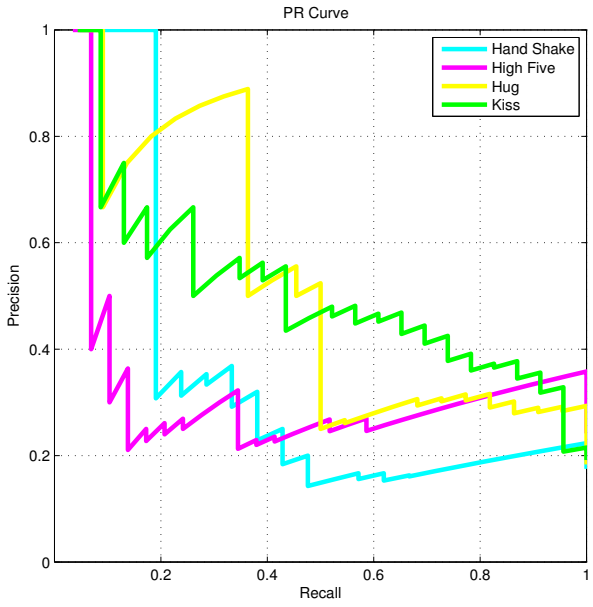

Figure 5.2: Per-class precision-recall curves for the TVShow dataset.

We also plot the confusion matrices for audio obtained with the best performance kernel configuration, (see the middle of Figure 5.5), as well as the per-class precision-recall curves (see Figure 5.2), those were obtained rom a retrieval experiment based on the BC kernel distance.

Finally, we present the classification accuracy obtained by using different kernels, and compared with results obtained from the BoW method. Table 5.5 and Table 5.6 show these comparisons. Our video trajectory consists of HOOF and MH features. It can be seen that the best results

Results on Video Based Interaction Recognition		
KSS / $K(y_t, y'_t)$	Linear	TL with RBF
Binet-Cauchy (BC)	50	60
RBF with BC kernel distace	51.66	**64.16**

Table 5.5: Video classification accuracy based on the KSS model

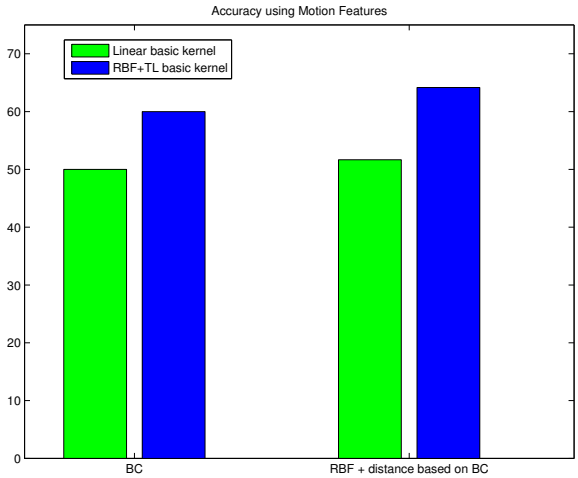

Figure 5.3: Classification accuracy summary based on visual features

Methods	Accuracy
Our Method	**64.16**
BoW	46

Table 5.6: Comparison between our method and the BoW approach for video interaction recognition.

have been obtained by using the RBF kernel with BC kernel distace and $K(y_t, y'_t)$ given by the tensor learning (TL) pairwise kernel. From table 5.6 it is apparent that our method performs better than the BoW method, indicating that the approach is promising.

5.2.3 Results on Audio-Video Based Interaction Recognition

In this section we present the results of the experiments conducted using audio and video features. We have used two types of video features given by the HOOF and MH. Whereas we use MFCC as audio features. As for the previous section we test dthe direct sum audio-video

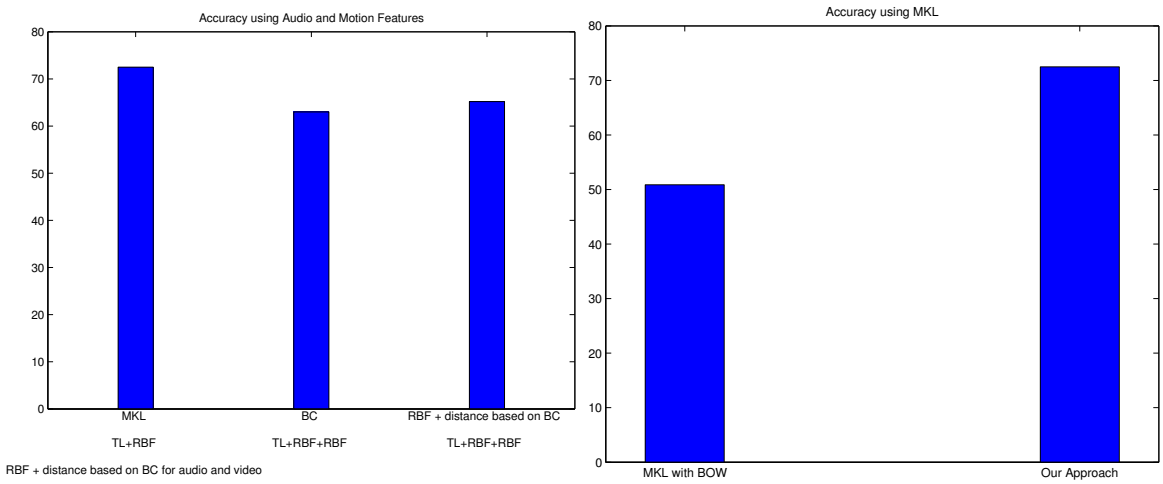

Figure 5.4: Classification accuracy of our methods (left), classification accuracy comparison between our approach and the BoW approach (right).

combination approach, and also the multiple kernel learning method. For the direct sum approach we tested the KSS model with a multi-class SVM classifier using the Binet-cauchy (BC) kernel. The audio part of the direct sum is using the RBF kernel, whereas the video part is using the tensor learning pairwise kernel. The same BC kernel is also used in another configuration, where it is converted into a BC kernel distance, and then used in combination with an RBF kernel. The results of the two configurations are presented in table 5.7. Also in table 5.7 are reported the results of audio and video combination based on the multiple kernel learning. In this case both the audio and video features exploit an RBF kernel with the corresponding BC kernel distance. The BC kernels for audio and video were computed in the same way they were for the direct sum approach. Figure 5.4 presents the classification accuracy among different approaches.

It can be seen from table 5.7 that the best results have been obtained by using the RBF kernel with kernel BC distace based, and $K(y_t, y'_t)$ given by a tensor learning (TL) pairwise kernel. Table 5.8 shows how our approach compares favorably against the BoW model. Also figure 5.5 shows the confusion matrices corresponding to the classification based only on audio cues (left), only video cues (center), and based on merging those with MKL (right).

Combined audioand video classification accuracy		
KSS / $K(y_t, y'_t)$	TL with RBF	TL with RBF
Binet-Cauchy (BC)	63	-
RBF with BC kernel distace	65.20	-
MKL	-	**72**

Table 5.7: Combined audio and video classification accuracy.

Methods	Accuracy
Our Method	**72**
BoW	50.86

Table 5.8: Comparison between our method and the BoW method when audio and video are combined.

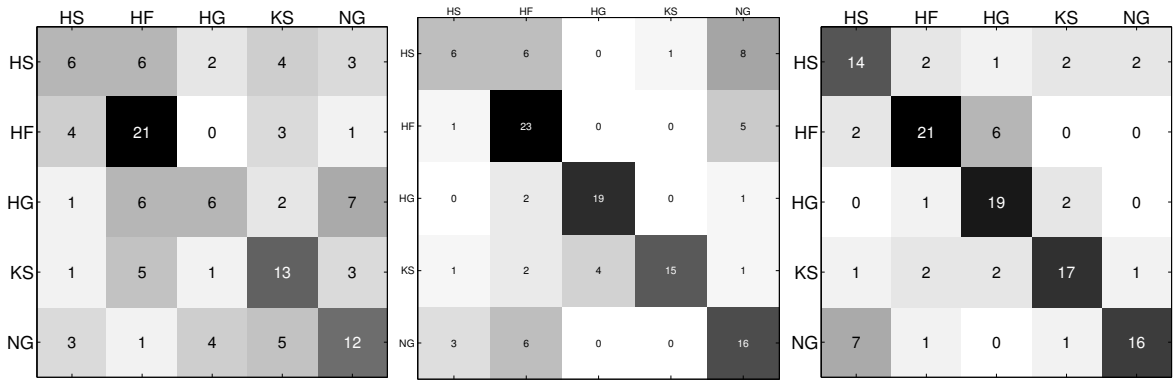

Figure 5.5: Confusion matrix for audio based classification (left), for video based classification (Middle), for combination based classification (right).

Chapter 6

Conclusion

In this thesis we propose a framework for modeling and recognizing binary human interactions, which is based on audio and visual features. We propose to model the visual information by a temporal sequence of motion features extracted from video, forming visual interaction trajectories. Similarly, we propose to model the audio information by a temporal sequence of audio features, synchronized with the motion features, and forming audio interaction trajectories. We develop a framework where visual and audio trajectories are modeed as the output of kernel state space (KSS) models. Therefore, recognizing audio and/or visual trajectories entails the ability to recognize KSS models. Such recognition can be supported by the use of a kernellized version of the recently proposed Binet-Cauchy kernels, which can be used for training multi-class SVM classifiers.

A crucial challenge addressed by the proposed framework is how to combine the information carried by the audio trajectory together with the information of the visual trajectory. To this end we propose two different approaches. The first one performs the direct sum of the audio and visual feature spaces, and exploits the KSS modeling framework to classify interactions. The second approach seeks for an optimal combination of the kernels for audio and visual information in a multiple kernel learning framework.

The proposed approaches were extensively tested on a dataset made of videos of TV shows and Hollywood movies. A comparison with the only other available approach, based on the Bag-of-Words model, has revealed that our newly developed framework clearly outperforms previous methods, and sets a new state-of-the-art in this particular application. Future developments of this approach will involve the inclusion of other proxemics cues, such as gaze, in order to further

improve the classification accuracy, and the testing on more human interaction datasets which inclue also audio information.

References

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

[2] "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.

[3] 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.

[4] 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.

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

[6] M. J. Marín-Jiménez R. Muñoz-Salinas E. Yeguas-Bolivar N. Pérez de la Blanca, "Human interaction categorization by using audio-visual cues," 2013.

[7] 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.

[8] 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.

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

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

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

[12] 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.

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

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

[15] 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.

[16] 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.

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

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

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

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

[27] 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.

[28] 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.

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

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

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

[38] 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.

[39] 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.

[40] Feiqi Deng1 Qiuxia Wu1;2 Qiuxia Wu-Zhiyong Dagan Feng Qiuxia Wu1;2, Zhiyong Wang2, "Realistic human action recognition with audio context," *IEEE Human Action Recognition*, p. 293, 2010.

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

[42] 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.

[43] 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.

[44] 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.

[45] S. Mermelstein Davis, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *In IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. Vol. 28 No., pp. 357–366, 1980.

[46] P. Davis, S. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980.

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

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

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

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

[51] 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.

[52] Rizwan Chaudhry and Rene Vidal, "Recognition of visual dynamical processes: Theory, kernels, and experimental evaluation," vol. 3400 N, pp. 35, 2099.

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

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

[55] 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.

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

[57] Mehmet Gonen and Ethem Alpaydn, "Multiple kernel learning algorithms," *Journal of Machine Learning Research*, vol. 12, no. 2211-2268, 2011.