2014

Online Geometric Human Interaction Segmentation and Recognition

Harika Bharthavarapu

Online Geometric Human Interaction Segmentation and Recognition

by

Harika Bharthavarapu

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
Electrical Engineering

Don Adjeroh, Ph.D.
Xin Li, Ph.D.
Gianfranco Doretto, Ph.D., Chair

Lane Department of Computer Science and Electrical Engineering

Morgantown, West Virginia
2014

Keywords: Temporal segmentation, Human Interaction recognition, Kernel Regression Model, Kernel State Space Model

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346

Abstract

Online Geometric Human Interaction Segmentation and Recognition

by

Harika Bharthavarapu
Master of Science in Electrical Engineering

West Virginia University

Gianfranco Doretto, Ph.D., Chair

The goal of this work is the temporal localization and recognition of binary people interactions in video. Human-human interaction detection is one of the core problems in video analysis. It has many applications such as in video surveillance, video search and retrieval, human-computer interaction, and behavior analysis for safety and security. Despite the sizeable literature in the area of activity and action modeling and recognition, the vast majority of the approaches make the assumption that the beginning and the end of the video portion containing the action or the activity of interest is known. In other words, while a significant effort has been placed on the recognition, the spatial and temporal localization of activities, i.e. the detection problem, has received considerably less attention. Even more so, if the detection has to be made in an online fashion, as opposed to offline. The latter condition is imposed by almost the totality of the state-of-the-art, which makes it intrinsically unsuited for real-time processing.

In this thesis, the problem of event localization and recognition is addressed in an online fashion. The main assumption is that an interaction, or an activity is modeled by a temporal sequence. One of the main challenges is the development of a modeling framework able to capture the complex variability of activities, described by high dimensional features. This is addressed by the combination of linear models with kernel methods. In particular, the parity space theory for detection, based on Euclidean geometry, is augmented to be able to work with kernels, through the use of geometric operators in Hilbert space. While this approach is general, here it is applied to the detection of human interactions. It is tested on a publicly available dataset and on a large and challenging, newly collected dataset. An extensive testing of the approach indicates that it sets a new state-of-the-art under several performance measures, and that it holds the promise to become an effective building block for the analysis in real-time of human behavior from video.

Acknowledgements

I would first like to thank my committee chair and advisor, Dr. Gianfranco Doretto, for giving me the opportunity to work with him and his students. This thesis would not be possible without his constant guidance and support.

I would also like to thank Dr. Donald Adjeroh and Dr. Xin Li for being on my committee. I have been fortunate to have had the opportunity to take courses with the committee members, and their teachings have been essential to my understanding of the subject.

Next, I would also like to thank the students in the Computer Vision research lab with whom I've had the pleasure of working alongside. In particular, I would like to thank my colleague Saeid Motiian, who has been a great help to me.

Finally, I would like to express my gratitude to my family. First to my dad, Prasad, for keeping me going through all the stresses and deadlines, while also dealing with his own. I would be remiss if I did not also express my thanks to my mother and sister. Their support seemingly has no limit, and has been much appreciated throughout my life.

Contents

List of Figures

List of Tables

Notation

We use the following notation and symbols throughout this thesis.

$\phi(\cdot)$	:	Mapping function
$\mathcal{S}$	:	Input feature space
$\mathcal{H}$	:	Hilbert space
$\{\cdot\}$	:	Temporal sequence
$\mathbb{H}$	:	Histogram space
$\mathbb{R}^n$	:	Real space with n dimension
v_t	:	System noise
w_t	:	Observation noise
λ	:	Weight
$\|\cdot\|$	:	Matrix norm
ν	:	Threshold
$\mathbf{y}_{i,j}$	:	Interaction trajectory of the persons i and j
κ	:	Kernel
$\mathbf{h}$	:	Histogram of oriented optical flow feature
$\mathbf{m}$	:	Motion Histogram
$(\cdot)^\top$	:	Transpose
$\doteq$	:	Approximately equal
τ	:	Test time window frames
τ_h and τ_m	:	number of optical flow and motion histogram bins respectively

Chapter 1

Introduction

Temporal segmentation of human interaction sequences (in 2D videos or surveillance videos) into segments with meaningful semantics, is an important step for building an intelligent framework to analyze human interactions. This temporal localization of events can be applied to content based video retrieval, human-object interfaces and video understanding and behavior analysis. Temporal localization is crucial for human action and simultaneously interaction recognition. The recognition of human activities is an important step towards the long-term goal of achieving a fully automatic understanding of scene, which in general characterized by the actions and interactions being performed by the people involved in it (e.g. *walking and approaching eachother for an hug or handshake and then departing in different directions*, etc.).

Recent works in human activity recognition focus was mostly limited to simple primitive actions like walking, running, and jumping. Thus, recognizing daily interactions like handshakes, hugging, etc., along with interactions like stabbing, shooting, etc., for security purposes, which are composed of complex temporal patterns, relies on accurate temporal structure decomposition [1]. Moreover most of the existing recognition frameworks assumes that the sequence is segmented to contain only the interaction part, leaving the space for event localization. Previous work on temporal segmentation was addressed mainly with statistical methods and clustering for unsupervised learning approaches. Many works in statistics, even the quickest change point detection [2] often works offline and restricted to simple 1D data or under the assumption that the distribution is known in advance [3]. Even

the temporal clustering proposed for unsupervised learning of human motions [4] is usually performed offline. Thus, because of the complex structure of motion dynamics, these approaches are not suitable for realtime video segmentation and interaction recognition. Hence, the need for temporally segmenting the videos online and analyzing them i.e recognizing the interactions (here) for high-dimensional represented data.

The main goal of this thesis is to address the above need for online temporal segmentation and recognition of human-human interactions in videos sequences. The main challenge is handling the complexity of the variability of data that represent human activities, which are inherently multidiensional. This has been handled by combining the representation in kernel Hilbert space with the use of the *parity space* in Hilbert space, and deriving closed from statistics based on kernel evaluations for online segmentation.

In this work we propose an online approach to cope with the high dimensions of the data, as well as the complexity of their variability by combining notions from two well understood theories and formalisms. The first one is the theory on reproducing kernel Hilbert spaces [5], and the second is the theory on state space models [6]. Exploiting the power of kernels allows a flexible and effective blending of heterogeneous high-dimensional features which can be mapped into a suitable Hilbert space where they can easily be modeled, even with linear models. Exploiting the theory on state space models allows borrowing a number of well understood results about their estimation, and their power for doing analysis, recognition, and detection based on multidimensional temporal sequences.

The resulting approach allows to extend the notion of *parity space*, developed within the context of detection based on linear models [7], for its use together with *kernel regression*, and *kernel state space* models, which are the Hilbert space counterparts of the linear versions. Rather than using Euclidean geometry to project data onto the parity space and reveal a detection, we exploit the geometry of linear operators in Hilbert space, and derive closed form solutions for the computation of normalized test statistics, based solely on kernel evaluations.

The framework based on kernel state space models allows to account for the temporal correlation of activities, and can easily be extended to do recognition [8, 9]. In particular, binary human interaction recognizing is addressed. These can be represented by temporal sequences, and require the use of pairwise kernels to model the symmetry of their space [10].

The recognition approach introduced in [9, 10], is combined with the temporal segmentation to obtain an online segmentation and recognition framework. This framework is suitable to work online through the use of a temporal incremental window, and online parameter estimation techniques, such as online kernel PCA [11, 12], and recursive least squares [6], through which realtime performance can be achieved.

To test this combined segmentation and recognition framework, we collected a new, large and challenging dataset of binary human interactions, along with the widely used state-of-the-art dataset have been proposed and used. Results are encouraging and supporting theory, showed better results for kernel state space models when compared similar approach Maximum Mean Discrepancy [13] for online temporal segmentation. The evaluation protocol of [14] has been used for comparison of three (*KSS, KR, MMD*) methods.

The detailed description of the proposed models: *KR and KSS* is available in Section § 3. Section § 4 show how the models can be used for segmenting and recognizing temporal sequences, and human binary interactions in particular. Finally, Section § 5 validates the proposed approach by achieving very promising results. Section § 2 discusses the previous related works on segmentation and recognition.

Chapter 2

Literature Review

2.1 Related work on Temporal Segmentation

Testing for change-point in a signal is an important problem that arise in many applications. Detecting potential changes can be either the final goal, as in surveillance and monitoring applications, or an intermediate step that is required to allow further processing an interpretation. Temporal segmentation is a multifaced area and a good amount of search have been done spreading for its applications, in statistics, computer vision and graphics, few of which are summarized in this section.

Change-Point Detection(CD)

In general, CD has been addressed in statistics, works better for univariate series i.e. one dimensional signals and where the parametric distribution assumptions are allowed, which does not hold for human activities with complex structure. The state of the art frameworks addressing CD though fast would work offline. Undirected sparse Gaussian graphical models along with jointly structured estimation and segmentation has also an explored area in CD [15]. Recent works [3] tried to extend CD as a non parametric Bayesian online change-point detection (BOCD). A regime to combine BOCD and Gaussian Process(GPs) are combined to relax the i.i.d for segmenting high complex human activities represented in a temporal sequence [16], although its framework of using GPs was successful in modeling complex data, it failed to keep computational cost low. Most recently the problem of modeling

high dimensional complex data is being addressed using kernel, where the data representation is being done in RKHS (Reproducible Kernel Hilbert Space). Kernel methods have been applied to non-parametric change-point detection on multivariant time series which is more relevant to the proposed framework of this thesis [17, 18]. Most of the kernel change-point detection (KCD) [17] utilizes one-class SVM as online training method and [18] performs segmentation based on the Kernel Fisher Discriminant Ratio. Most similar work on these lines has been done in [13] where online temporal segmentation has been addressed using incremental or growing window which resets to prefixed window size after every detected cut or change-point and it used MMD for comparing the distributions from the partitioned window for human activities. In this thesis though we have tried both incremental and sliding window, incremental window has showed better results for binary human interactions and we compared over MMD as in [13] where no training is needed for segmentation.

Temporal Clustering

The problem of temporal segmentation has been addressed using clustering concept as in machine learning [19, 20] and extended to correctly temporally segment time series into different clusters. For this temporal clustering, a combination of kernel K-means and spectral clustering, Aligned Cluster Analysis (ACA) have been used though not for motion analysis or segmentation but for facial expression change or for facial behavior monitoring using a multi-subject correspondence algorithm for matching facial expressions [21]. The switch linear dynamical system (SLDS) has been improved after solving the problem of estimating unknown number of clusters using the hierarchical Dirichlet process [22]. Unsupervised approaches to modeling and segmentation include entropy minimization to construct Hidden Markov Models (HMMs), with high level behaviors mapped to states of the HMM [23]. HMMs and other clustering techniques require an expensive search process and a very good initial guess of the parameters [24]. As we can see that these clustering works offline where labeling of clusters is provided as in clustering (predefined). Temporal segmentation proposed in this thesis works online and is suitable for real time applications. Though we also have clustering models based on distance metrics developed over a variety of temporal scales [25]

and online segmentation based on consideration of information loss [26],they can only handle a wide range of dynamic situations (like outdoor and indoor scenes changes) and may not work for surveillance applications, in this work interaction detection which involve minute motion change detection.

Motion Analysis

Grouping human motions by motion analysis is one of many ways that the problem of segmentation has been addressed in computer vision and graphics. Graph spectral clustering has been used to focus works on unusual human activity detection [27]. Spatio-temporal features also have been used to address event clustering in video sequences [28]. Geometric-invariant temporal clustering algorithm has been proposed in [4] to cluster facial expressions. From the data mining community subspace clustering [29] was of particular interest. This approach is designed to identify low-dimensional clusters in high-dimensional data. However, this framework requires to consider motion as unordered set of poses. Using this framework with Gaussian Mixture Models proved that it is easier to locally capture a transition between two behaviors than the clustering of unordered poses.

More relevantly, an elegant temporal extension of Probabilistic Principal Component Analysis for change-point detection (PPCA-CD) with an online algorithm to decompose motion sequences into distinct action segments [30]. Their work also included PCA and GMM models for motion segmentation among which PPCA proved to have better results. PCA-based methods for motion segmentation showed better results only for the data modeled by Gaussian clouds. For GMM, it was needed that all sequences in the database contain approximately the same number of behaviors (keeping # clusters constant) this might not be the case for real time video segmentation. PPCA proved perform well, though computationally efficient, restricted to (approximate) Gaussian assumptions. Also, their experiments were restricted to motion capture data which is very simple when compared to video data involving high-dimensional complex data. Hence, we have used kernel regression models for our complexly challenging data.

A simple approach for temporal segmentation is to reduce it to sequence of test for homo-

geneity between two parts of a sliding window over the signal to perform change detection. As they scale linearly in the length of the signal, they are known for their scalability, such approaches are attractive when compared to retrospective approaches taking the signal as a whole and typically scale quadratically in the length of the signal [31, 32]. The main characteristics of the data we consider lies in its high-dimensional feature vector. Hence, classical parametric multivariant test statistics cannot be applied [33], which produce very low detection rate because availability of few samples to estimate high-dimensional quantities appearing in the test statistics. A promising nonparametric alternative to parametric approaches is offered by kernel-based methods.

2.2 Activity Recognition Synopsis

Human action and activity recognition is of significant interest in applications that range from computer game development to public security monitoring. This technology of human action and activity recognition was developed and inspired by object recognition techniques. From video complexity, action recognition can be divided into single person action recognition, human to human interaction (also called as binary interaction) recognition, and group activity recognition. In this thesis we mainly focused on binary human interactions.

Oliver et al. [34] constructed a variant of the basic HMM, the coupled HMM, to model human-human interactions. The major limitation of the basic HMM is its inability to represent activities composed of motions of two or more agents. A HMM is a sequential model and only one state is activated at a time, preventing it from modeling the activities of multiple agents. They introduced the concept of Coupled HMM to model complex interactions between two persons. It is constructed by coupling two HMMs to model human-human interactions. More specifically, they coupled the hidden states of two different HMMs by specifying their dependencies. Their system was able to recognize complex interactions between two persons, such as concatenation of 'two persons approaching, meeting, and continuing together'.

Around 2004, J.K. Aggarwal's research group in university of Texas at Austin developed a hierarchical method for binary interaction recognition [35, 36]. They divided human motion

to body part movements such as Torso's movement and arm's movement. According to head pose information and body parts information, they classified the interaction to different categories. With a new realistic dataset, this research group developed a video structure comparison method in later years [37]. This well-known new dataset is called as UT-dataset. So far, it is still the most popular dataset for binary interaction study. In their work, they extracted histogram based spatio-temporal local features from videos. After that, they create a match kernel which belong to Mercer's kernel and use this match kernel to measure the similarity of feature structures from different videos. Then they localize the detected atom activity by searching the activity's spatial coordinates, starting time, and ending time which is based on voting. Through hierarchical recognition, the detected binary interaction can be classified. With this system, more complicated binary interactions are able to be recognized. Compared with previous works, the approach proposed in their work greatly improve the recognition accuracy for the realistic binary interaction.

In 2012, Patron-Perez et al. developed a new approach to recognize binary interactions in video from their new TVHI dataset [38]. They tracked all upper bodies and heads in a video and developed a person centered descriptor based on the head orientations and the local spatio-temporal region around them. From the information of local cues, they obtained the spatial relationship between people and head orientations, which are called as global cues. Then they use structure SVM to learn and inference on their model to obtain the interaction class. Besides their new dataset, they also performed their model on UT dataset. The classification accuracy is even better than that of Brendel' work.

Structured learning has been used for several applications in Computer Vision. Blaschko and Lampert [39] used it to learn a mapping between images and object bounding boxes to model context information. Desai [40] used structured SVM to learn spatial relations between object categories aiming to obtain a simultaneous classification of all bounding boxes in the image, while Wang [41] use it to learn both dependencies between objects and object attributes, and between the attributes themselves.

With a new BIT interaction dataset, another approach was proposed by [42]. They used high-level descriptions, which is called interactive phrases, to represent binary semantic motion relationships between those interacting people. These motion relationships between

arms, legs, and torsos could be leg stepping forward, arm stretching, static torso, and etc. And they treated these interactive phrases as latent variables. Finally, they classify the interaction types by using latent SVM. They tested their model on both BIT and UT datasets and got encouraging results.

This thesis, uses temporal interaction trajectories coupled together with the body motion of each individual as well as their proximity relationships to model binary people interactions. Such trajectories are modeled with a non-linear dynamical system (NLDS). Framework that entails the use of so-called pairwise kernels, to compare interaction trajectories in the space of NLDS has been used [10]. This work also include modeling the Riemannian structure of the trajectory space, and kernels satisfying symmetric property which are peculiar of interaction modeling.

Chapter 3

Online Temporal Segmentation Models

In this section the problem of segmenting a time series online is addressed. We then apply the approach to the recognition of human interactions. We deploy a temporal sliding window [24, 13], and sequentially detect segmentation cuts.

Monitoring a temporal sequence $\{\mathbf{y}_t\}$, assume the last segmentation cut was observed at time $s < t$, where t is the current time. We want to test whether at time $t - \tau$ a new cut should be detected. To this end either a kernel regression model (3.1), or a kernel state space model (3.9) is estimated from the data in the *training time window* $[s+1, \cdots, t-\tau]$, of length $T_t \doteq t - \tau - s$, i.e. $\mathbf{y}_{s+1}, \cdots, \mathbf{y}_{t-\tau}$. See Figure 3. A cut should be detected if the data observed in the subsequent *test time window* $[t-\tau+1, \cdots, t]$, i.e. $\mathbf{y}_{t-\tau+1}, \cdots, \mathbf{y}_t$, and

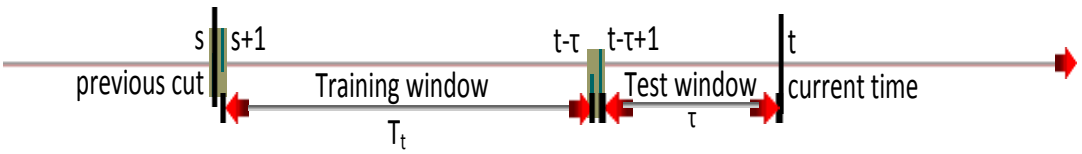

Figure 3.1: Sliding Window

the model previously estimated do not fit "well enough".

To determine whether the two distributions - the test time window distribution and the training time window distribution are equal to not to decide on a cut. Two famous statistical tests [5] to determine if the new sample or set of samples in a window (τ) are from same distribution are discussed. The problem of temporal segmentation can be generalized as the problem of deciding the similarity or dissimilarity of the distributions. In application areas like bio-informatics, where the interest lies in comparison of micro-array data taken from different tissue types, either to determine whether two subtypes of cancer may be treated as statistically indistinguishable from a diagnosis perspective, or to detect differences in healthy and cancerous tissue and in database attribute matching, where it is desirable to merge databases containing multiple fields without the prior information of their field correspondence: the fields are matched by maximizing the similarity in the distributions of their entries.

3.1 Kernel Regression Model

Let $\{\mathbf{y}_t\}$ be the input temporal sequence on which segmentation has to be performed. It lies in a space $\mathcal{S}$ (may not be Euclidean). This sequence have to be mapped onto feature space $\mathcal{H}$(Hilbert Space) for further processing. We use Mercer Kernel $\kappa(\mathbf{y}_t, \mathbf{y}'_t) = \langle \phi(\mathbf{y}_t), \phi(\mathbf{y}'_t) \rangle$ to obtain the mapping: $\mathcal{S} \to \mathcal{H}$. If we consider that the sequence $\{\mathbf{y}_t\}$ is mapped to $\{\phi(\mathbf{y}_t)\}$, then the *kernel regression(KR) model* is given by

$$\phi(\mathbf{y}_t) = C\mathbf{x}_t + w_t \tag{3.1}$$

Here, C is a linear operator $C : \mathbb{R}^n \to \mathcal{H}$ represented as $C \doteq [c_1, \cdots, c_n]$, $\mathbf{x}_t \in \mathbb{R}^n$ represented as $\mathbf{x} \doteq [x_1, \cdots, x_n]^\top$ and $C\mathbf{x} \doteq \sum_{i=1}^{n} c_i x_i$. The observation noise w_t is modeled as a zero-mean Gaussian process.

Assuming that the temporal sequence $\{\mathbf{y}_t\}$ is made of i.i.d samples and is modeled by (3.1), input sample $\mathbf{y}_t$ is measured for its accordance with the model. This can be done with the help of *kernel parity vector* ξ_t, as it indicates the direction and magnitude of the sample deviation from the span of $\{c_i\}$. It is given as $\xi_t \doteq P_{\mathcal{P}}\phi(\mathbf{y}_t)$ and also as $\xi_t = P_{\mathcal{P}} w_t$,

where $P_{\mathcal{P}}$ is an operator that projects a vector $v \in \mathcal{H}$ onto $\mathcal{P}$ ($P_{\mathcal{P}}v$). This is based on the concept of *kernel parity Hilbert space (KPHS)*, which is the subspace of $\mathcal{H}$ defined as $\mathcal{P} \doteq \{v \in \mathcal{H} | \langle c_i, v \rangle = 0, i = 1, \cdots, n\}$ [7].

Hence, ξ_t gives us the information of the measure of noise the new input sample $\mathbf{y}_t$ in feature space $\mathcal{H}$ into the parity space $\mathcal{P}$. Thus using ξ_t and knowing the noise model, we can decide whether the current sample $\mathbf{y}_t$ implies a noise model different than the one given or not.

Now, the *residual error* $e_t \doteq \phi(\mathbf{y}_t) - C\hat{\mathbf{x}}_t$, where $\hat{\mathbf{x}}_t$ is the maximum likelihood estimation of the regressor, given the observation $\mathbf{y}_t$, and the model given by κ and C.

3.1.1 Model Derivation

The parameters are estimated under the hypothesis of the noise w_t being i.i.d. realizations from an uncorrelated stationary Gaussian process, which means that its autocorrelation function is given by $\sigma^2\delta$, where δ is a Dirac distribution defined over a suitable domain, the maximum likelihood estimation $\hat{\mathbf{x}}_t$ coincides with the least squares estimation

$$\hat{\mathbf{x}}_t = \arg\min_{\mathbf{x}} \|\phi(\mathbf{y}_t) - C\mathbf{x}\|^2 . \tag{3.2}$$

To establish a rule to determine whether the sample $\mathbf{y}_t$ is in accordance with the model (3.1), we need to connect the residual error to the kernel parity vector along with the model estimation from the kernel κ and the samples $\mathbf{y}_1, \cdots, \mathbf{y}_T$. For that, we apply Kernel PCA (KPCA) [5] to model the variability of $\{\mathbf{y}_t\}$ in feature space.

Now, the kernel matrix is given as $K \doteq \Phi^\top\Phi$, where $\Phi \doteq [\phi(\mathbf{y}_1), \cdots, \phi(\mathbf{y}_T)]$ for convenience. After assuring that data in feature space has zero-mean, the linear combination coefficients are computed from the eigen decomposition of K: $JKJ \doteq \alpha\Lambda\alpha^\top$, where $\Lambda \doteq \text{diag}(\lambda_1, \cdots, \lambda_T)$ and α are the eigenvalue and eigen vector matrices, $J \doteq (I - \frac{1}{T}\mathbf{e}\mathbf{e}^\top)$ (centering projection matrix) and $\mathbf{e} = [1, \cdots, 1]^\top$($T$ here indicates evaluated kernel principal components out of a linear combination of the elements of ΦJ). In order to model the highest amount of data variability in feature space with only n components, we pick first n. Hence, the observation operator of the model is set to

$$\hat{C} \doteq \Phi J\beta . \tag{3.3}$$

where, $\beta \doteq \alpha \Lambda_n^{-\frac{1}{2}}$ (removing the columns of Λ after the first n).

The sample estimation for the noise model (from KPCA) is given by

$$\hat{\sigma}^2 = \frac{1}{T} \sum_{t=1}^{T} \|w_t\|^2 \tag{3.4}$$

To relate residual error with kernel parity vector, plug-in (3.3) in (3.2), removing the mean of the model $\frac{1}{T}\Phi\mathbf{e}$ and simplifying further(expand the square, derive w.r.t $\mathbf{x}$ and equating to zero) we obtain

$$\hat{\mathbf{x}}_t = \beta^\top J \breve{\kappa}(\mathbf{y}_t) \ , \tag{3.5}$$

where $\breve{\kappa}(\mathbf{y}_t) \doteq \left(\tilde{\kappa}(\mathbf{y}_t) - \frac{1}{T} K\mathbf{e}\right)$, and $\tilde{\kappa}(\cdot) \doteq [\kappa(\mathbf{y}_1, \cdot), \cdots, \kappa(\mathbf{y}_T, \cdot)]^\top$. Moreover, by combining (3.5) and (3.2) we can see that $\min_{\mathbf{x}} \|\phi(\mathbf{y}_t) - \frac{1}{T}\Phi\mathbf{e} - C\mathbf{x}\|^2 = \|P_{C^\perp}\left(\phi(\mathbf{y}_t) - \frac{1}{T}\Phi\mathbf{e}\right)\|^2$, where $P_{C^\perp}$ is the projection operator defined by

$$P_{C^\perp} = I - \Phi J \beta \beta^\top J \Phi^\top \ , \tag{3.6}$$

where I here indicates the identity operator. Thus, we can say that $e_t = P_{C^\perp}\left(\phi(\mathbf{y}_t) - \frac{1}{T}\Phi\mathbf{e}\right)$, and by construction $P_{C^\perp}$ represent an orthonormal projection onto the orthogonal complement of the span of the $\{c_i\}$, and therefore it is equivalent to $P_{\mathcal{P}}$.

Finally, $\|e_t\|^2 = \|\xi_t\|^2$. Thus, a simple check for establishing whether or not the new sample y_t is in accordance with model (3.1) is to verify if the *normalized residual error* $\|e_t\|^2/\sigma^2$ is lower or greater than a threshold ν, appropriately chosen.

3.1.2 Temporal Segmentation

The geometric framework above, project the test data onto the kernel parity Hilbert space (KPHS) and compare this projection with the noise model to decide whether data and model can fit. More formally, for the KR model one should compute the following statistic

$$\varepsilon_{t-\tau}^{KR} \doteq \frac{1}{\tau\sigma^2} \sum_{i=0}^{\tau} \|e_{t-i}\|^2 \ , \tag{3.7}$$

Finally, $\varepsilon_{t-\tau}^{KR}$ can be used to test the hypotheses "yes cut", i.e. $\mathbf{H}_1$, versus "no cut", i.e. $\mathbf{H}_0$. In particular,

$$\varepsilon_{t-\tau} \leq \nu \Rightarrow \mathbf{H}_0 \text{ is true}, \quad \varepsilon_{t-\tau} > \nu \Rightarrow \mathbf{H}_1 \text{ is true} \ . \tag{3.8}$$

If $\mathbf{H}_0$ is true, test (3.15) is repeated at time $t + \Delta t$. If $\mathbf{H}_1$ is true, the next test is performed at time $t + \tau$, with a training time window that restarts with length $T_{t+\tau} = \tau$.

3.2 Kernel State Space Model

For *kernel state-space (KSS)* model the regressor temporal/sate sequence is given by

$$\begin{cases} \mathbf{x}_{t+1} = A\mathbf{x}_t + v_t \ , \\ \phi(\mathbf{y}_t) = C\mathbf{x}_t + w_t \ . \end{cases} \tag{3.9}$$

Since, we considered that the samples of the temporal sequence $\{\mathbf{y}_t\}$ are correlated rather than i.i.d. as in the case of KR model. Here the dynamics of the state evolution is described by the new elements of the model are $A \in \mathbb{R}^{n\times n}$, and the zero-mean i.i.d. Gaussian distributed system noise v_t (covariance Q, independent from w_t).

Here we are trying to predict the model for the test window of size τ(t is current time). The segmentation is done based on the reconstruction error which is the measure of deviation of the actual model of test window from the predicted model. To formulate the reconstruction error we consider the following vector and matrix representations:

$\Phi^t_{t-\tau+1} \doteq [\phi(\mathbf{y}_{t-\tau+1})^\top, \cdots, \phi(\mathbf{y}_t)^\top]^\top$

$W^t_{t-\tau+1} \doteq [w^\top_{t-\tau+1}, \cdots, w^\top_t]^\top$

$\mathbf{V}^t_{t-\tau+1} \doteq [v^\top_{t-\tau+1}, \cdots, v^\top_t]^\top$

$$\tilde{O}_\tau \doteq \begin{bmatrix} 0 & \cdots & \cdots & 0 \\ & \ddots & \cdots & \vdots \\ & & 0 & \vdots \\ O_{\tau-1} & \cdots & O_1 & 0 \end{bmatrix} \tag{3.10}$$

As in the case of previous model, the concept known as observability matrix in the theory of linear dynamical systems (LDS) has been considered. Specifically, the linear operator $O_\tau : \mathbb{R}^n \to \mathcal{H}^\tau$, mapping x to $O_\tau\mathbf{x}$, where $O_\tau \doteq [C^\top, A^\top C^\top, \cdots A^{\tau-1\top} C^\top]^\top$ is taken into consideration. This extends the definition of KPHS into *kernel parity Hilbert space of order τ (KPHS-τ)*, which is the subspace of $\mathcal{H}^\tau$ defined as $\mathcal{P}_\tau \doteq \{v \in \mathcal{H}^\tau | v^\top O_\tau = 0\}$. Here, $\tilde{W}^t_{t-\tau+1} \doteq \tilde{O}_\tau \mathbf{V}^t_{t-\tau+1} + W^t_{t-\tau+1}$ is a zero-mean Gaussian process noise with autocorrelation matrix function $\tilde{O}_\tau I_\tau \otimes Q\tilde{O}^\top_\tau + I_\tau \otimes \sigma^2\delta$ ($\otimes$ indicates the Kronecker product).

Therefore, the reconstruction error is given as follows

$$E^t_{t-\tau+1} \doteq \Phi^t_{t-\tau+1} - O_\tau \hat{\mathbf{x}}_{t-\tau+1} \tag{3.11}$$

where $\hat{\mathbf{x}}_{t-\tau+1}$ is the maximum likelihood estimation of $\mathbf{x}_{t-\tau+1}$ and $\Phi^t_{t-\tau+1} = O_\tau \mathbf{x}_{t-\tau+1} + \tilde{W}^t_{t-\tau+1}$.

3.2.1 Model Derivation

The maximum likelihood estimation of $\mathbf{x}_{t-\tau+1}$ under few simplifying assumptions (like the auto-correlation matrix function of $\tilde{W}^t_{t-\tau+1}$ is given by $I_\tau \otimes \sigma^2\delta$) turns out to be a simple least squares estimation as

$$\hat{\mathbf{x}}_{t-\tau+1} = \left(\sum_{i=0}^{\tau-1} {A^i}^\top A^i\right)^{-1} \sum_{i=0}^{\tau-1} {A^{\tau-1-i}}^\top \beta^\top J\breve{\kappa}(\mathbf{y}_{t-i}) \, . \tag{3.12}$$

If we consider the projection operator to be

$$P_{O^\perp_\tau} \doteq I - O_\tau \left(\sum_{i=0}^{\tau-1} {A^i}^\top A^i\right)^{-1} O_\tau^\top \, . \tag{3.13}$$

we can re-write the reconstruction error as $E^t_{t-\tau+1} = P_{O^\perp_\tau}(\Phi^t_{t-\tau+1} - \mathbf{e}_\tau \otimes \frac{1}{T}\Phi\mathbf{e})$, where $\mathbf{e}_\tau$ is a column vector with τ ones.

$P_{O^\perp_\tau}$ represents an orthonormal projection onto the orthogonal complement of the span of the columns of O_τ, and therefore it is equivalent to $P_{\mathcal{P}_\tau}$-the operator that projects a vector $v \in \mathcal{H}^\tau$ onto $\mathcal{P}_\tau$, given by $P_{\mathcal{P}_\tau} v$, whereas $\Xi^t_{t-\tau+1} \doteq P_{\mathcal{P}_\tau}\Phi^t_{t-\tau+1}$ is the kernel parity vector. From the definition of KPHS-τ, $\Xi^t_{t-\tau+1} = P_{\mathcal{P}_\tau}\tilde{W}^t_{t-\tau+1}$, which shows that it is independent from the state $x_{t-\tau+1}$, and it can be interpreted with respect to $\mathbf{y}_{t-\tau+1}, \cdots, \mathbf{y}_t$ exactly in the same way as ξ_t is interpreted with respect to $\mathbf{y}_t$.

Hence, $\|E^t_{t-\tau+1}\|^2 = \|\Xi^t_{t-\tau+1}\|^2$ (under the hypothesis of $\tilde{W}^t_{t-\tau+1}$ being an uncorrelated stationary Gaussian process-an idealized scenario).

Similar to the KR model, the criterion for establishing whether or not the trajectory $\mathbf{y}_{t-\tau+1}, \cdots, \mathbf{y}_t$ is in accordance with model (3.9) is to simply check if the *normalized residual error* $\|E^t_{t-\tau+1}\|^2/\tau\sigma^2$ is lower or greater than a threshold ν, appropriately chosen.

Algorithm: Learning a kernel dynamic texture
Input: Video Sequence $[y_1 \cdots, y_N]$, state space dimension n, kernel function $\kappa(y_1, y_2)$
Compute the mean: $\overline{y} = \frac{1}{N}\sum_{i=t}^{N} y_t$
Subtract the mean: $y_t \leftarrow y_t - \overline{y}, \forall_t$
Compute the (centered) kernel matrix $[K]_{i,j} = \kappa(y_i, y_j)$
Compute KPCA weights α from K
$[\widehat{x}_1 \cdots \widehat{x}_N] = \alpha^T K$
$\widehat{A} = [\widehat{x}_2 \cdots \widehat{x}_N][\widehat{x}_1 \cdots \widehat{x}_{N-1}]^\dagger$
$\widehat{v}_t = \widehat{x}_t - \widehat{A}\widehat{x}_{t-1}, \forall t$
$\widehat{Q} = \frac{1}{N-1}\sum_{t=1}^{N-1} \widehat{v}_t \widehat{v}_t^T$

Table 3.1: Computing A and Q

3.2.2 Temporal Segmentation

From the above geometric framework projecting the test data onto the kernel parity Hilbert space (KPHS-τ), and then comparing the projection with the noise model to decide whether data fit the model. So, for the KSS model one should compute

$$\varepsilon_{t-\tau}^{KSS} \doteq \frac{1}{\tau\sigma^2}\|E_{t-\tau+1}^{t}\|^2 \ . \tag{3.14}$$

Finally, $\varepsilon_{t-\tau}^{KSS}$ can be used to test the hypotheses "yes cut", i.e. $\mathbf{H}_1$, versus "no cut", i.e. $\mathbf{H}_0$. In particular,

$$\varepsilon_{t-\tau} \leq \nu \Rightarrow \mathbf{H}_0 \text{ is true}, \quad \varepsilon_{t-\tau} > \nu \Rightarrow \mathbf{H}_1 \text{ is true} \ . \tag{3.15}$$

If $\mathbf{H}_0$ is true, test (3.15) is repeated at time $t + \Delta t$. If $\mathbf{H}_1$ is true, the next test is performed at time $t + \tau$, with a training time window that restarts with length $T_{t+\tau} = \tau$.

3.3 Online Model Parameter Estimation

The kernel PCA is the kernelized version of standard PCA. With standard PCA, the data is projected on to the linear subspace that best captures the variability of the data. In contrast kernel PCA (KPCA) projects the data on to non-linear functions in the input-space. These non-linear principal components are defined by the kernel function. For the KSS and KR models when we consider training window of particular length and testing window of variable length, the model parameter matrix A is computed as shown in the table .

The estimates of the state variable are the KPCA coefficients and the state-space parameters can be estimated with the least squares method. A non-linear dynamical system learned in this manner is called a kernel dynamic texture because it uses KPCA to learn the state space variables, rather than PCA as with the standard dynamic texture. The learning algorithm is summarized in which self explanatory.

For the approach to work online, the online kernel PCA algorithm in [43] and the matrix A is recursively updated online as explained in [44], which is summarized briefly in this section.

The test window size would be one frame for the online approach, if the training window $[\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_N]$ of length N with the kernel matrix K_N is what we have and have to calculate the kernel matrix associated with the new test data point added to the training points, $[\mathbf{y}_2, \mathbf{y}_3, \cdots, \mathbf{y}_{N+1}]$ denoted as K. This updating is done using the Update method of De Moor. For the update we assume that we have the dominant NxM eigenspace U_{NM} of the square symmetrical matrix K_N and we suppose that an estimate for the number of dominant eigenvectors M is available.

In the kernel matrix an update is located in the last row and column, which expands K_N both in row and column dimension when a point is added $K = \begin{bmatrix} K_N & \mathbf{a} \\ \mathbf{a}^T & b \end{bmatrix}$ where $\mathbf{a}$ is a Nx1 vector of kernel entries $a_i = \kappa(\mathbf{y}_i, \mathbf{y}_{N+1})$ and scalar $b = \kappa(\mathbf{y}_{N+1}, \mathbf{y}_{N+1})$.

The key observation is that K can be expanded as a stripped matrix to which two rank-1 matrices are added.The possible convenient expansion could be:

$$K = \begin{bmatrix} K_N & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix} + \frac{1}{2} \begin{bmatrix} \mathbf{a} \\ \frac{b}{2} + 1 \end{bmatrix} \begin{bmatrix} \mathbf{a} \\ \frac{b}{2} + 1 \end{bmatrix} \begin{bmatrix} \mathbf{a}^T & \frac{b}{2} + 1 \end{bmatrix} - \frac{1}{2} \begin{bmatrix} \mathbf{a} \\ \frac{b}{2} - 1 \end{bmatrix} \begin{bmatrix} \mathbf{a}^T & \frac{b}{2} - 1 \end{bmatrix} \tag{3.16}$$

where $\mathbf{0}$ is Nx1 vector of zeros. Let us denote the given rank-M SVD of the submatrix $K_N \approx U_{NM} \Sigma_{NM} (U_{NM})^T$. We can then organize the separate vectors all together in a convenient symmetrical factorization. With the rearrangement we might obtain an eigenspace, $U_0 = \begin{bmatrix} U_{NM} \\ \mathbf{0}^T \end{bmatrix}$ with a set of orthonormal eigenvectors in U_0, extended with the two columns of B ($B := \begin{bmatrix} \mathbf{a} & \mathbf{a} \\ \frac{b}{2} + 1 & \frac{b}{2} - 1 \end{bmatrix}$), giving the matrix $\begin{bmatrix} U_0 & B \end{bmatrix}$ The two column vectors of B disturb the orthogonality of this matrix. To restore this orthogonality, we need to compute the part

of B that is orthogonal to the eigenspace. So we must decompose the vectors of A into a component orthogonal and a component parallel to the U_0. This splits up the extending contribution of A into an orthogonal component that increases the rank of the eigenspace, while the parallel component will cause the eigenvectors to be rotated. This leads to a factorization. To make the column vectors of orthogonal component mutually orthogonal, we compute the Q R-decomposition. In order to obtain the optimal eigenvectors, a small SVD on the three middle matrices is necessary. The smallest singular value and corresponding eigenvector will then be discarded, which gives $K \approx U'_{NM}\Sigma'_M(U'_{NM})^T$.

Update Algorithm:

$$U_0 = \begin{bmatrix} U_{NM} \\ \mathbf{0}^T \end{bmatrix}$$

$$Q_B R_B \xleftarrow{\text{QR}} (I - U_0 U_0^T) \begin{bmatrix} \mathbf{a} & \mathbf{a} \\ \frac{b}{2}+1 & \frac{b}{2}-1 \end{bmatrix}$$

$$Q_u = \begin{bmatrix} U_0 & Q_B \end{bmatrix}$$

$$\Sigma_u = R_u \begin{bmatrix} \Sigma_m & \mathbf{0} \\ \mathbf{0}^T & D_u \end{bmatrix} R_u^T$$

Hence we obtain $K \approx Q_u R_u \begin{bmatrix} \Sigma_m & \mathbf{0} \\ \mathbf{0}^T & D_u \end{bmatrix} R_u^T Q_u^T$.

Now to obtain the matrix A we use the adaptive online parameter estimation [44]. It is an efficient and recursive parameter estimation procedure that is also adaptive, because it gives more importance to recent measurements according to a forgetting factor, and allows the detection system for adjusting to slow variations of the visual process, favoring the reduction of false alarms.

It is possible to update the state matrix A, after the update of C given the state $\mathbf{x}_t$. This is done with a variant of the recursive least squares algorithm [6], which leads to the following update equations

$$L(t) = \frac{\mathbf{x}_{t-1}^T \Sigma(t-1)}{\lambda + \mathbf{x}_{t-1}^T \Sigma(t-1)\mathbf{x}_{t-1}}$$

$$\hat{A}(t) = \hat{A}(t-1) + (\mathbf{x}_t - \hat{A}(t-1)\mathbf{x}_{t-1})L(t)$$

$\Sigma(t) = \frac{1}{\lambda}\Sigma(t-1)(I - \mathbf{x}_{t-1}L(t))$

where $L(t)$ is the Kalman gain, $\Sigma(t)$ is a covariance matrix, and I here is the identity matrix. λ is forgetting factor that exponentially weights the measurements by giving more importance to the recent ones.

3.4 Maximum Mean Discrepancy

If p and q are considered to be the two distributions from the training time and test time windows respectively, they are to be tested for similarity based on which MMD segmentation is performed. On the basis of samples drawn from each of the distributions, a smooth function which is large on the points drawn from p and small(as less as possible, may be negatives) on the points from q. The difference between the mean function values on the two samples will be the test statistics. When the distance exceeds the threshold set, it is likely that the samples are from different distributions, from which we can say that the distributions are different or dissimilar. This statistics is known as Maximum Mean Discrepancy (MMD).

The accuracy of MMD in determining the similarity of the distributions depend highly on the class $\mathcal{F}$ of smooth functions used. $\mathcal{F}$ must be balanced between the two criterion to be *rich enough* to vanish the MMD population if and only if p=q and it must be *restrictive enough* for the empirical estimate of MMD to converge quickly to its expectation as the sample size increases. MMD is known to be computationally cheap: given m points sampled from p and n from q, the cost is $O(m+n)^2$ time.

3.4.1 MMD Derivation

The derivation of empirical estimate of MMD [45] goes as follows

The Two-Sample-Problem

Statistical test formulated should be able to satisfy the following conditions:

If p and q are distributions defined on a domain $\mathcal{X}$. Given observations $X := \{x_1, ...x_m\}$ and $Y := \{y_1, ...y_n\}$ (i.i.d.) drawn from p and q respectively, then $p \neq q$.

Based on the fact that the two distributions are equal if and only if their Borel probability measures $(Ep(f(x)), Eq(f(x)))$are equal, we can define the criterion that the MMD takes a unique high value when $p = q$.

To determine a function class $\mathcal{F}$, that uniquely allows $C(\mathcal{X})$ to identify $p = q$ is not practical in the finite sample setting. Thus unspecified more general class $\mathcal{F}$, is defined to measure the discrepancy between p and q as proposed in [45].

Then MMD and its empirical estimate in more general form-

$$MMD[F, p, q] := \sup_{f \epsilon F}(E_{z\sim p}[f(x)] - E_{y\sim q}[f(y)]) \tag{3.17}$$

$$MMD[\mathcal{F}, p, q] := \sup_{f \epsilon F}(\frac{1}{m}\sum_{i=1}^{m} f(x_1) - \frac{1}{n}\sum_{i=1}^{n} f(y_1)) \tag{3.18}$$

Now, the challenging part is to identify a function class that is rich enough to establish $p = q$ uniquely, yet restrictive enough to provide useful finite sample estimates. For simplicity we consider $\mathcal{F}$ to be the unit ball in universal RKHS $\mathcal{H}$ [46]. As mentioned in [46] the universal kernels Gaussian and Laplace kernels as used.

To make MMD easy to compute, we consider the fact that in an RKHS, function evaluations can be written $f(x) = \langle \phi(x), f \rangle$, where $\phi(x) = \kappa(x, .)$. Denote by $\mu[p] := E_x \epsilon p(x)[\phi(x)]$ the expectation of $\phi(x)$, assuming that $\| \mu[p] \|^{\frac{2}{4}} < \infty$ is a sufficient condition which can be arranged for our comfortability as $E_p[\kappa(x, x')] < \infty$, where x and x' are independent random variables drawn according to p. From $E_p[f(x)] = \langle \mu[p], f \rangle$ we have

$$MMD[\mathcal{F}, p, q] = \sup_{\|f\|_{\mathcal{H}} \leq 1} \langle \mu[p] - \mu[q], f \rangle = \| \mu[p] - \mu[q] \|_{\mathcal{H}} \tag{3.19}$$

Using $\mu[X] := \frac{1}{m}\sum_{i=1}^{m} \phi(x_1)$ and $\kappa(x, x') = \langle \phi(x), \phi(x') \rangle$, an empirical estimate of MMD is

$$MMD[\mathcal{F}, X, Y] = \Big[\frac{1}{m^2}\sum_{i,j=1}^{m} \kappa(x_i, x_j) - \frac{2}{mn}\sum_{i,j=1}^{m,n} \kappa(x_i, x_j) + \frac{1}{n^2}\sum_{i,j=1}^{n} \kappa(y_i, y_j)\Big]^{\frac{1}{2}} \tag{3.20}$$

Hence, the above equation provides us with a test statistic for $p \neq q$. Although the above estimate is straight forward to upper bound, it is biased further. For now we expect MMD[$\mathcal{F}$,p,q] to be small if $p = q$, and the quantity to be large if the distributions are far apart.

3.4.2 Hypothesis Testing

The next most important part in determining the similarity of the i.i.d distributions is to describe a framework of statistical hypothesis testing. The statistical test, $\mathcal{T}(X, Y) : \mathcal{X}^m \times \mathcal{X}^n \to \{0, 1\}$ where $X \sim p$ of size m and $Y \sim q$ of size n, is used to distinguish between the null hypothesis $\mathcal{H}_0 : p = q$ and the alternative hypothesis $\mathcal{H}_1 : p \neq q$. This is achieved by setting a particular threshold: if it exceeds threshold, the distributions are said to be different rejecting the null hypothesis since, zero population of MMD indicates $p = q$. Thus any real number below the threshold falls under the acceptance region of the test.

Uniform Convergence Bounds

The two essential properties of MMD pave the way for determining a unique convergence bound for the given set of data. First property: the empirical MMD converges in probability at rate $(m + n)^{-1/2}$ to its population value. Second property: for large deviations of the empirical MMD in the case $p = q$ probabilistic bounds are given.The threshold for the hypothesis we are using can be lead by these bounds.

$$MMD[\mathcal{F}, X, Y] > m^{-\frac{1}{2}} \sqrt{2E_p[\kappa(x, x) - \kappa(x, x')]} + \epsilon > 2(K/m)^{\frac{1}{2}} + \epsilon \tag{3.21}$$

[47]

where, K is assumed to be the upper bound of the kernel $|\kappa(x, y)| \leq K$ (this notation K is different from the one used in the previous section). From the above equation, two possible bounds can be illustrated. $B_1(\mathcal{F}, p)$ and $B_2(\mathcal{F}, p)$ on the basis in the empirical estimate (3.21). $B_1(\mathcal{F}, p)$ links between the bias bound and kernel size. Hence this alone is not sufficient. Thus, $B_2(\mathcal{F}, p)$ is used to bound the bias, based the fact that, a hypothesis test of level α for the null hypothesis $p = q$ (equivalently MMD[$\mathcal{F}$,p,q]=0) has the acceptance region $MMD[\mathcal{F}, X, Y] < 2\sqrt{K/m}(1 + \sqrt{\log \alpha^{-1}})$. Here, the test statistics has $n + m$ - 2 degrees of freedom, and its error probability converges at the same rate as our hypothesis test. [α acts like the threshold in the previous sections which has to be appropriately chosen).

The selection of kernel parameters is another important issue in the practical application of MMD-based tests. This can be illustrated with a Gaussian RBF kernel, where we must choose the kernel width σ. Both for kernel size $\sigma = 0$ and $\sigma \to \infty$, the empirical MMD is

zero. As a heuristic compromise between the two extremes, the aggregate sample median is set to be σ. However, the optimum choice of kernel size is an ongoing research.

3.5 Chapter Summary

The theory of reproducing Kernel Hilbert spaces and the theory on state space models is combined to represent the highly complex high dimensional data. Exploiting the power of kernels allows a flexible and effective blending of heterogeneous high-dimensional features which can be mapped into a suitable Hilbert space where they can easily be modeled, even with linear model. Exploiting the theory on state space models allows borrowing a number of well understood results about their estimation, and their power of doing analysis, recognition, and detection based on multidimensional temporal sequences.

Two main detection approaches - *Kernel Regression and Kernel State Space* models which are the Hilbert space counterparts of the linear versions are discussed. The geometry of Hilbert space linear operators is exploited and closed form solutions for the computation of normalized test statistics, based only on kernel evaluations are derived. The accuracy of these models is compared with the well known MMD technique.

KSS model is proved to be the best performer among the three as it should be, supporting the theory.

Kernel Regression (KR) Model:

The test data is projected on to the kernel parity Hilbert space(KPHS) and this projection is compared with noise model to decide the data fit the model. Residual error is computed as shown below, based which a segmentation cut is declared.

$$\varepsilon_{t-\tau}^{KR} \doteq \frac{1}{\tau\sigma^2} \sum_{i=0}^{\tau} \|e_{t-i}\|^2 \lessgtr \nu \tag{3.22}$$

Kernel State Space (KSS) Model:

The training data projected on to the kernel parity Hilbert space (KPHS$-\tau$)is used to predict the model. Reconstruction error is calculated as given below, upon which a

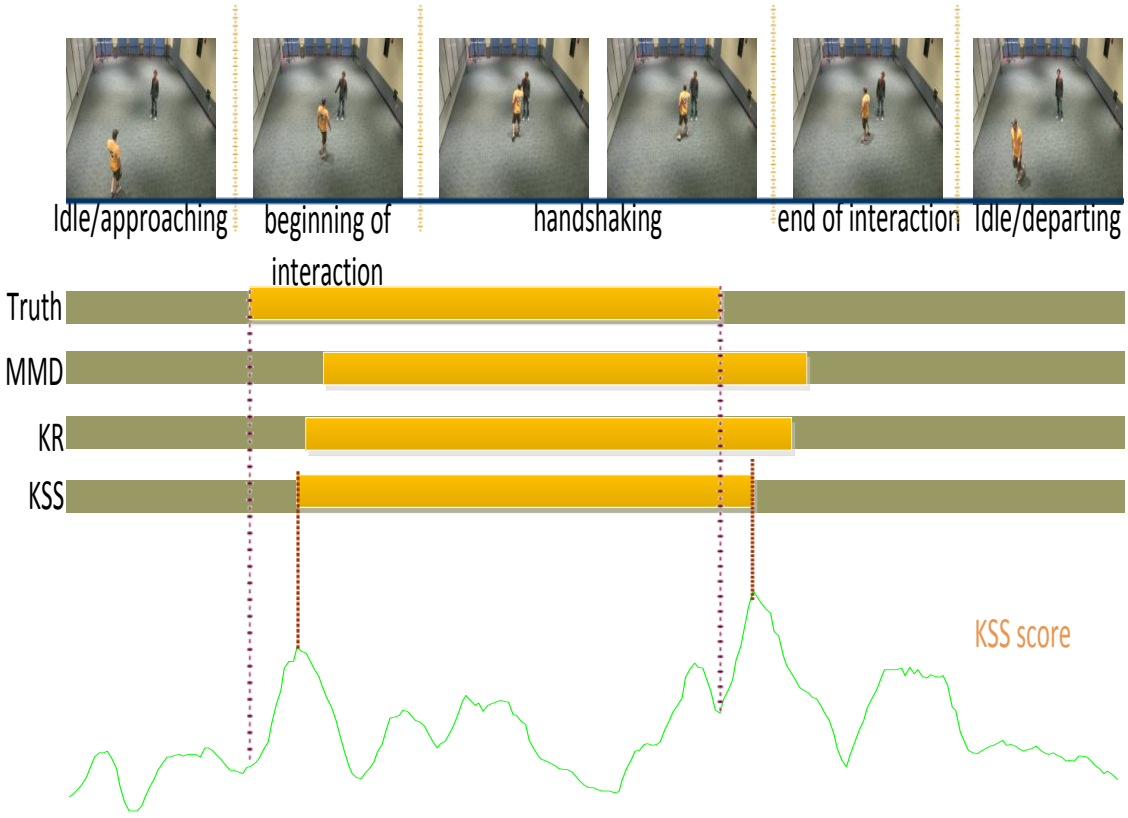

Figure 3.2: KSS Score for a typical interaction

segmentation cut is noted.

$$\varepsilon_{t-\tau}^{KSS} \doteq \frac{1}{\tau\sigma^2}\|E_{t-\tau+1}^{t}\|^2 \leqslant \nu \tag{3.23}$$

MMD:

The above two frameworks proposed newly are compared with the model know to be successful in temporal segmentation based on kernels. The distance between the means of two distributions is computed as below to decide on the cut.

$$MMD[\mathcal{F}, X, Y] \leqslant 2\sqrt{K/m}(1+\sqrt{\log \alpha^{-1}}) \tag{3.24}$$

Chapter 4

Interaction Recognition

4.1 Human Interaction Recognition

In this section we model interaction trajectories as the output of non-linear dynamical systems (NLDS), and reduce the problem of recognizing human interactions to the problem of discriminating between NLDSs. This involves designing special kernels that satisfy both the geometry of the space where the interaction trajectories live, and certain symmetry properties, which are induced by the fact that we are modeling binary people interactions. Both the constraints are satisfied by carefully exploiting kernel construction techniques, and by clearly showing that kernels for recognizing interaction trajectories should belong to a subcategory of the so-called *pairwise kernels*, and in particular they should satisfy the *balanced property*, which not only boost the performance but also reduces the training time avoiding the use of symmetric dataset, which would be double the size of a regular one.

4.1.1 Feature Extraction

As we are particularly concerned with recognizing binary interaction in a video, at every frame the bounding box delimiting the region of each person is assumed to be given (through the use of our annotation tool, like it is typically done in video surveillance settings). For the i-th person in the video sequence, at every time t the bounding box is used to extract features aiming at describing the body motion.

From each bounding box two features - *histogram of oriented optical flow* (HOOF) [9](the

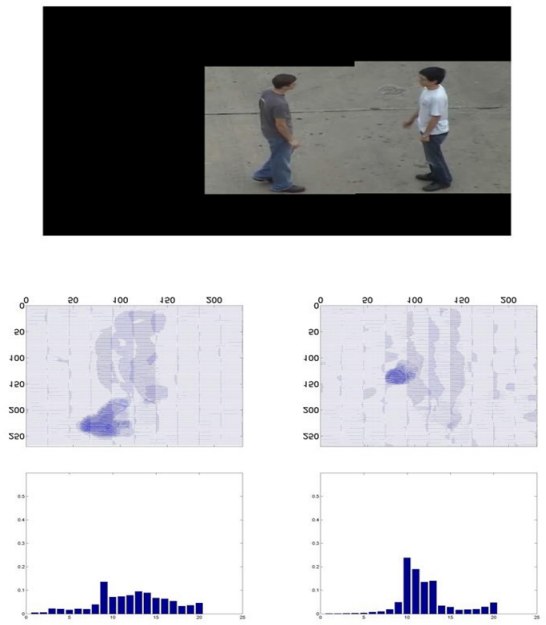

Figure 4.1: An example of HOOF descriptor. a) Binary interaction image cut from video. b) Optical flow of left person. c) Optical flow of right person. d) histogram bins obtained from b. e) histogram bins obtained from c.

motion between two consecutive frames) and *motion histogram* (MH)(motion trajectory of the past $\tau - 1$ frames) are computed. The i-th person is represented by the sequence of HOOF and MH features $h_i \doteq \{h_{i,t}\}_{t=1}^{T}$, and $m_i \doteq \{m_{i,t}\}_{t=1}^{T}$, respectively, where $h_{i,t}$ and $m_{i,t}$ are normalized histograms made of τ_h bins, $h_{i,t} \doteq [h_{i,t;1}, \cdots, h_{i,t;\tau_h}]^\top$, and made of τ_m bins, $m_{i,t} \doteq [m_{i,t;0}, m_{i,t;1}, \cdots, m_{i,t;\tau_m-1}]^\top$ (bin 0 has been added to account for the case of absence of motion).

The brief description of extraction of HOOF and MH features is as follows:

HOOF: Optical flow, as one of the methods to detect human motion, is defined as apparent visual motion and the changes of light in the scene. The second row of Figure 4.1 shows an example of optical flow image. However, optical flow detection is susceptible to the variation of scales, background noise, and the direction of movement. To overcome these problems, HOOF, based on the distribution of optical flow, was proposed by Chaudhry et al. in 2009. They binned the flow vector though its angle and magnitude weight and then normalized the histogram. This makes HOOF independent of direction

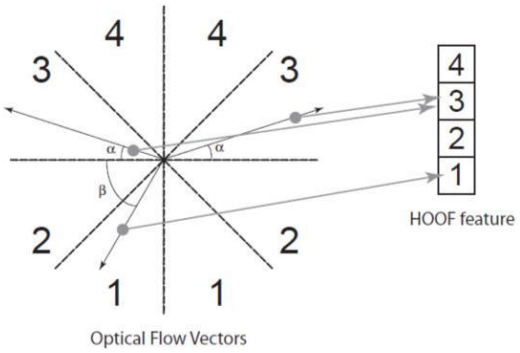

Figure 4.2: Histogram formation with 4 bins

of motion and scale variation. The third row of Figure 4.1 shows the histogram bins obtained from the optical flow images, and Figure 4.2 shows how histogram was formed in this method. From Figure 4.2, HOOF is symmetry in the orientation of the optical flow which indicates this feature is independent of direction of motion.

MH: This feature summarizes the motion trajectory of the past $\tau - 1$ frames (where $\tau > 1$). To obtain MH, we first need to compute the *motion image*, $M_t \doteq \sum_{k=1}^{\tau-1} \eta(I_t - I_{t-k})$, where $\eta(z) = 1$ if $|z| < \delta$, otherwise $\eta(z) = 0$. Here δ is a threshold parameter to be set. Once the motion image is computed, it was used to bin inside the bounding box of person to obtain the motion histogram of person i at frame t, $\mathbf{m}_{i,t}$. MH features are also scale invariant, robust to noise, and independent of direction. Figure 4.3 shows a couple of examples of motion images with the corresponding MH features. Here, vertical axis is normalized histogram and horizontal axis is the number of bins.

Based on the fact the two interacting persons have to be close enough like for *handshake, hugging*, etc., interactions[though we also have included *stabbing, shooting, waving* interactions for which distance might not be small], distance between the interacting persons is also taken into consideration. This information is captured by the Euclidean distance between the position $p_{i,t}$ of person i, and the position $p_{j,t}$ of person j, given by

$$d_{i,j,t} \doteq \|p_{i,t} - p_{j,t}\|_2 \,. \tag{4.1}$$

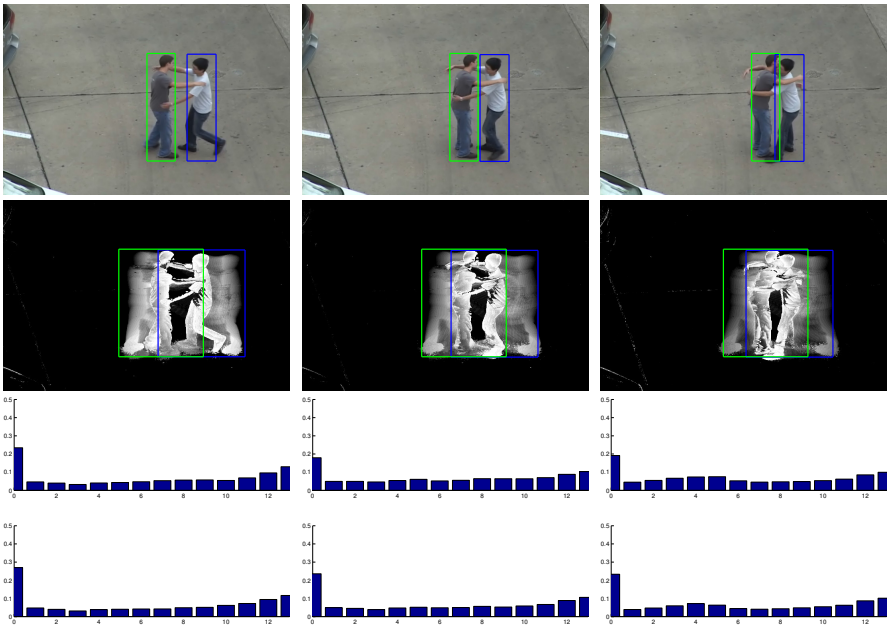

Figure 4.3: Motion images and MH feature trajectories (UTI)

When the camera calibration is known and people tracking is performed on the ground-plane, the person position and velocity are readily available. If this is not the case, one can characterize proximity by computing the distance in the image domain, and performing a normalization based on the people size. Even if doing so is not view invariant, our experimental results show that this information still significantly increases the classification accuracy for the tested datasets.

Given the motion, described by (h_i, m_i) and (h_j, m_j), of person i and j, and their proximity described by $d_{ij} \doteq \{d_{ij,t}\}$, their *interaction trajectory* is the temporal sequence $y_{ij} \doteq \{y_{ij,t}\}_{t=1}^{T}$, where

$$y_{ij,t} \doteq [h_{i,t}^{\top}, m_{i,t}^{\top}, h_{j,t}^{\top}, m_{j,t}^{\top}, d_{ij,t}]^{\top} \ . \tag{4.2}$$

.

Simplifying the representation of the features of the trajectories of persons i and j, we get $y_t \doteq [i_t, j_t, d_t]^{\top}$. As it is obvious that the interaction between persons (i, j) should be the same as the interaction between persons (i, j) in a typical video, so-called pairwise kernels [48] that account for this special symmetry, as well as for the geometric structure of the input space $\mathcal{S}$ (non-Euclidean) are used. In particular, we used for KR and KSS

modeling (reported to be the best performer [48]).

$$\kappa((i,j,d),(i',j',d')) = \kappa^{TL}((i,j),(i',j'))e^{-\gamma(d-d')^2} \tag{4.3}$$

where κ^{TL} is the tensor learning pairwise kernel, constant γ is estimated with cross-validation. And for MMD model we used simple RBF kernel (reported to be the best performer [13])

$$\kappa((i,j,d)) = \kappa^{RBF}(i,j)e^{-\gamma(d)^2} \tag{4.4}$$

4.1.2 Recognition

The temporal sequence segment $\mathbf{Y}_{s:t} \doteq [\mathbf{y}_s, \cdots, \mathbf{y}_t]$, obtained from the online segmentation containing binary human interactions, characterized by a temporally correlated sequence is to be recognized. Therefore, recognizing a segment entails comparing KSS models. When $\mathcal{S}$ is a non-Euclidean space, where the temporal sequence is assumed to lie, it is possible to compare KSS models through the use of Binet-Cauchy kernels [49]. In particular, [9] describes their use for action recognition when the input features are a temporal sequence of histograms, and [50] uses them for modeling and recognizing binary temporal sequences. Since our implementation framework is based on the features above [10], we apply the Binet-Cauchy kernel that they refer to as κ_{NLDS} and which embeds TL kernel.

A family of Binet-Cauchy kernels for LDSs extended to NLDSs like the KSS system model is used for recognition. In particular, the Binet-Cauchy trace kernel for NLDS is the expected value of an infinite series of weighted inner products between the outputs after embedding them into the high-dimensional (possibly infinite) space using the map $\phi(\cdot)$. More precisely

$$\kappa_{NLDS}(\{\mathbf{y}_t\}_{t=1}^{\infty}, \{\mathbf{y}'_t\}_{t=1}^{\infty}) \doteq E\left[\sum_{t=1}^{\infty} \lambda^t \phi(\mathbf{y}_t)^\top \phi(\mathbf{y}'_t)\right] = E\left[\sum_{t=1}^{\infty} \lambda^t \kappa(\mathbf{y}_t, \mathbf{y}'_t)\right] , \tag{4.5}$$

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 (4.5) can be computed in closed form, and it requires the computation of the infinite sum

$$P = \sum_{t=1}^{\infty} \lambda^t (A^T)^\top F A'^\top , \tag{4.6}$$

where $F = \tilde{\alpha} S \tilde{\alpha}'$, and the columns of $\tilde{\alpha}$ and $\tilde{\alpha}'$ are the centered KPCA weight vectors of $\{\mathbf{y}_t\}$ and $\{\mathbf{y}'_t\}$, given by $\tilde{\alpha}_c = \alpha_c - \frac{\mathbf{e}^\top \alpha_c}{T}\mathbf{e}$, and $\tilde{\alpha}'_d = \alpha'_d - \frac{\mathbf{e}^\top \alpha'_d}{T'}\mathbf{e}$, respectively. S instead is

such that $[S]_{st} = \kappa(\mathbf{y}_s, \mathbf{y}'_t)$, where $s \in \{1, \cdots, T\}$, and $t \in \{1, \cdots, T'\}$. If $\lambda \|A\| \|A'\| < 1$, where $\|\cdot\|$ is a matrix norm, then P can be computed by solving the corresponding Sylvester equation $P = \lambda A^\top P A' + F$.

Given P, kernel (4.5) can be computed in closed form provided that the co-variances of the system noise, the observation noise, and the initial state are available. On the other hand, like [9] 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 (4.5) is simplified to κ^{σ}_{NLDS}, which is a kernel only on the dynamics of the NLDS, and is given by the maximum singular value of P, i.e.,

$$\kappa^{\sigma}_{NLDS} = \max \sigma(P) \ . \tag{4.7}$$

We have also used a new kernel inspired from the concept of MMD, from κ_{NLDS} we form a Gaussian kernel based on the derived kernel distance, which we found to be more effective, and that is given by

$$\kappa^{KSS}(\mathbf{Y}, \mathbf{Y}') = e^{-\eta(\kappa_{NLDS}(\mathbf{Y},\mathbf{Y}) + \kappa_{NLDS}(\mathbf{Y}',\mathbf{Y}') - 2\kappa_{NLDS}(\mathbf{Y},\mathbf{Y}'))} \tag{4.8}$$

With the above kernel we use the libSVM [51] to train a multiclass SVM classifier.

4.2 Chapter Summary

The temporal sequence segment obtained from the online segmentation, under the form of a time series, coupled together with body motion of each individual along with their proximity relationships, are modeled with a non-linear dynamical system (NLDS). The framework uses pairwise kernels which are able to compare the interaction trajectories in the space of NLDS. Kernels peculiar of interaction modeling framework, proved to satisfy certain symmetry properties are chosen and the Riemannian structure is modeled to address the problem of binary interaction recognition [10].

Chapter 5

Results

5.1 Experimented Datasets

The implemented framework was experimented on the following datasets of varying complexity.

5.1.1 UT-Interaction Dataset

We worked on five human-human interaction classes: handshake, hug, kick, punch and push available in this dataset. This dataset includes two sets, divided based on their complexity. We experimented on both the sets and managed to obtain better results comparatively. Sequences are segmented to eliminate no-interaction time usually observed at the beginning of the videos. VATIC annotation tool is used to draw bounding boxes around the persons involved in the interaction. As, we could get the ground-truth suiting our feature extraction from the provided one we annotated the original dataset. Camera and Viewpoints are kept constant throughout the data.

5.1.2 HAUS-PI Dataset

We derived the subset of interactions from the large set of human-human interactions available at Human Activity Under Surveillance Dataset. For our experimental set we considered 12 PIs (Person Interaction Classes): handshake, hugging, high-five, kicking, punching,

pushing, slapping, waving, shooting, stabbing, patting. The length of the dataset along the with sequences length make this dataset more challenging. As, in this we observe people entering into scene walk from different directions to interact with the other person in the scene-view variance makes the difference of this dataset from the other most widely used datasets for this interaction detection and recognition purpose. Sequence length is reasonably good keeping the interaction for nearly 2-3 seconds at the center capturing the time for the persons to enter and leave the scene. The dataset is ground-truth with our annotation tool. Having a corresponding projection matrix for the ground plane extraction loaded, the tool drops a bounding box of required dimensions at one click. Spline interpolation between the tracks allows the user to save a lot of time as the trajectory looks more likely. Specifying the attributes for the labeled bounding boxes, saving and retrieving the annotated files is user friendly and computation efficient compared to the other annotation tools we experimented.

Besides PIs from HAUS dataset indoor lab settings, PIs are also been collected in the outdoor environment. The huge HAUS dataset not only includes PIs but also group actions where 2 or 3 groups performing various actions while 1 or 2 groups act like non participants. We have also collected PIs both indoor and outdoor scenarios in cluttered scenes where we see non participants moving around the scene. These sub datasets are more complicated when compared to just the ones we selected as they add more disturbance to the tracked scene, and hence we have not included them in the preliminary test data for our approach. The varying light and wind in the outdoor setting added more shadow effect to the scenes and made the entire setting complex when compared to the indoor setting because of which we kept it for future test dataset of our model.

5.2 Temporal Segmentation results

Segmentation results using MMD, the KR and the KSS models in comparison with the ground-truth annotations, for 7 randomly selected sequences from 12 classes of the HAUS-PI dataset can be seen the figure5.2. In this figure, each set of bars correspond to the interaction label given at the top. Each set include bars representing their corresponding active part in for the given interaction sequence. The results from various methods are compared for the

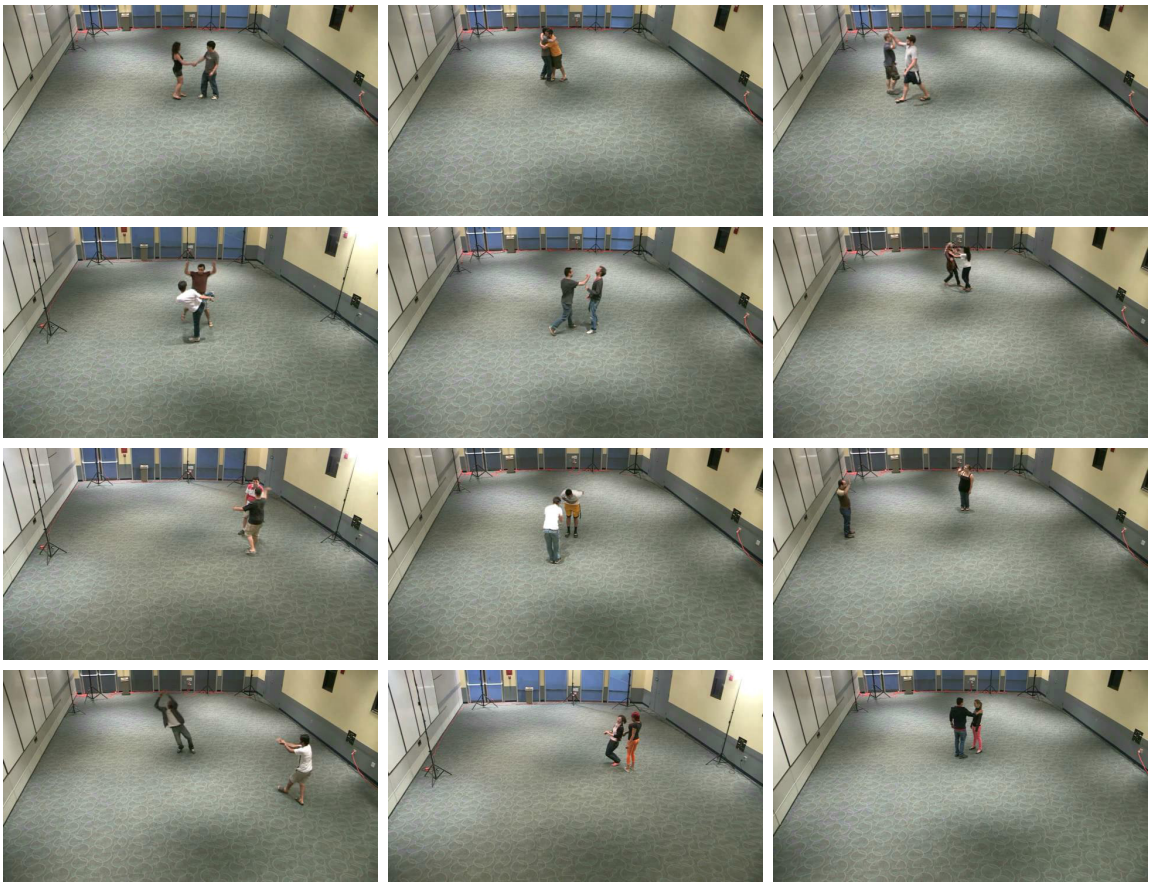

Figure 5.1: HAUS-PI

best parameter setting(threshold, test window length, etc.,).

Observing above figure, one can say that the start of interaction is well extracted by KSS model when compared to other two models, that is KSS model is able to detect the change with a tolerable delay (few frames from the ground-truth detection). We can also draw that the KSS model keeps the interaction area or the overlap area between the ground-truth active frames and its own detected active frames reasonably good, by balancing the total number of active frames to recognize the type of interaction. Though KR and MMD fired change-point detection soon after KSS model MMD shows poorer interaction end point detection, resulting in adding noise frames (frames containing no-interaction, like departing) to classification. Hence we can conclude that the KSS model shows better performance, as expected, for most of the interactions.

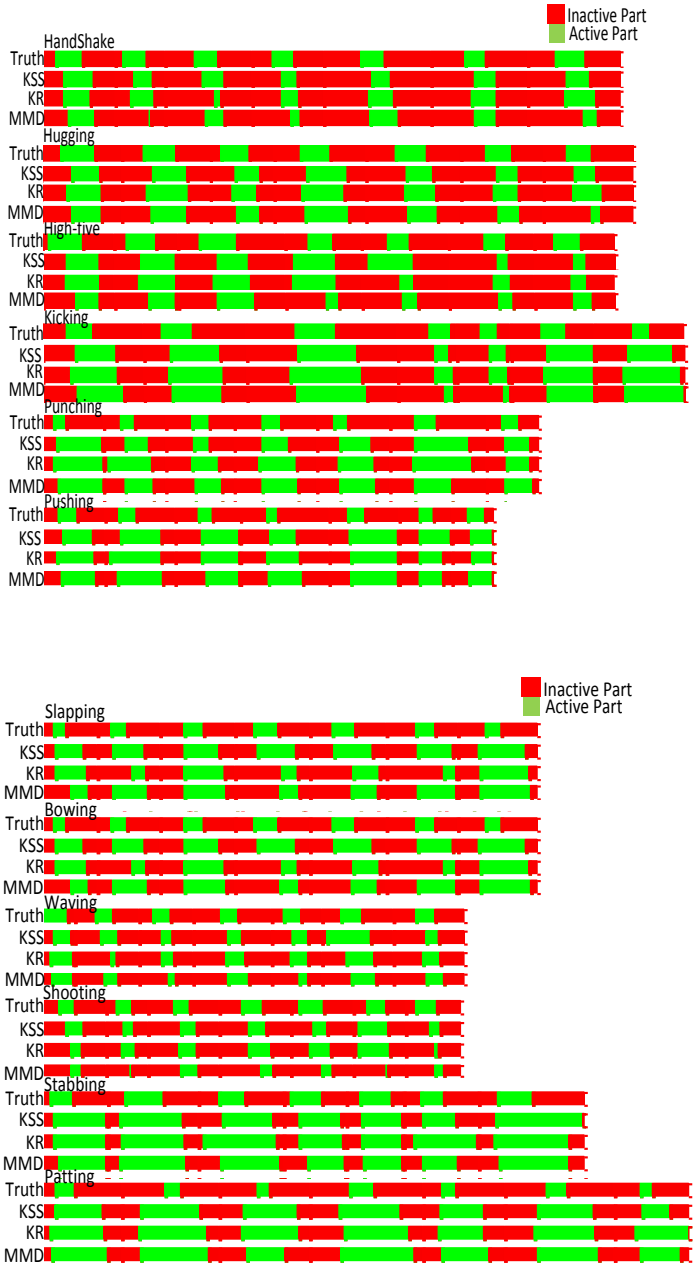

Figure 5.2: Temporal Segmentation results

5.3 Timeliness Accuracy

For evaluating the timeliness of detection we use Normalized Time to Detection (NTtoD) as a benchmark measure.

5.3.1 FPR

In statistics, when performing multiple comparisons, the term false positive ratio, usually refers to the probability of falsely rejecting the null hypothesis for a particular test. The false positive rate usually refers to the expectancy of the false positive ratio. Here, the False Positive Rate of the detector is defined as the fraction of frames that the detector fires before the event of interest starts.

5.3.2 TPR

True positive rate also called the Sensitivity, measures the proportion of actual positives which are correctly identified as such (e.g. the percentage of sick people who are correctly identified as having the condition). Here, the True Positive Rate is defined as the fraction of frames that the detector fires during the event of interest.

5.3.3 NTtoD

If the interaction is active in the interval [a,b] and the interaction offset is detected at time t, then

$$NTtoD = t - a + 1/(b - a + 1) \tag{5.1}$$

NTtoD = 0 if t¡a and NTtoD = infinity for a false rejection, i.e., t¿b.

5.3.4 AMOC

Activity Monitoring Operating Curves are plots drawn between NTtoD and FPR. They are obtained by varying the change score detection threshold, while keeping the other parameters set at the optimal value. Sensitivity of the normalized time to detection with respect to the length τ of the test time window, for the MMD model (left), for the KR model (center) and for the KSS model (right). Below centered figure is the graph comparison between the KSS, KR, and MMD models. We observe that, though MMD performance varies with τ, other two models showed a stable performance for different values of test time window. In

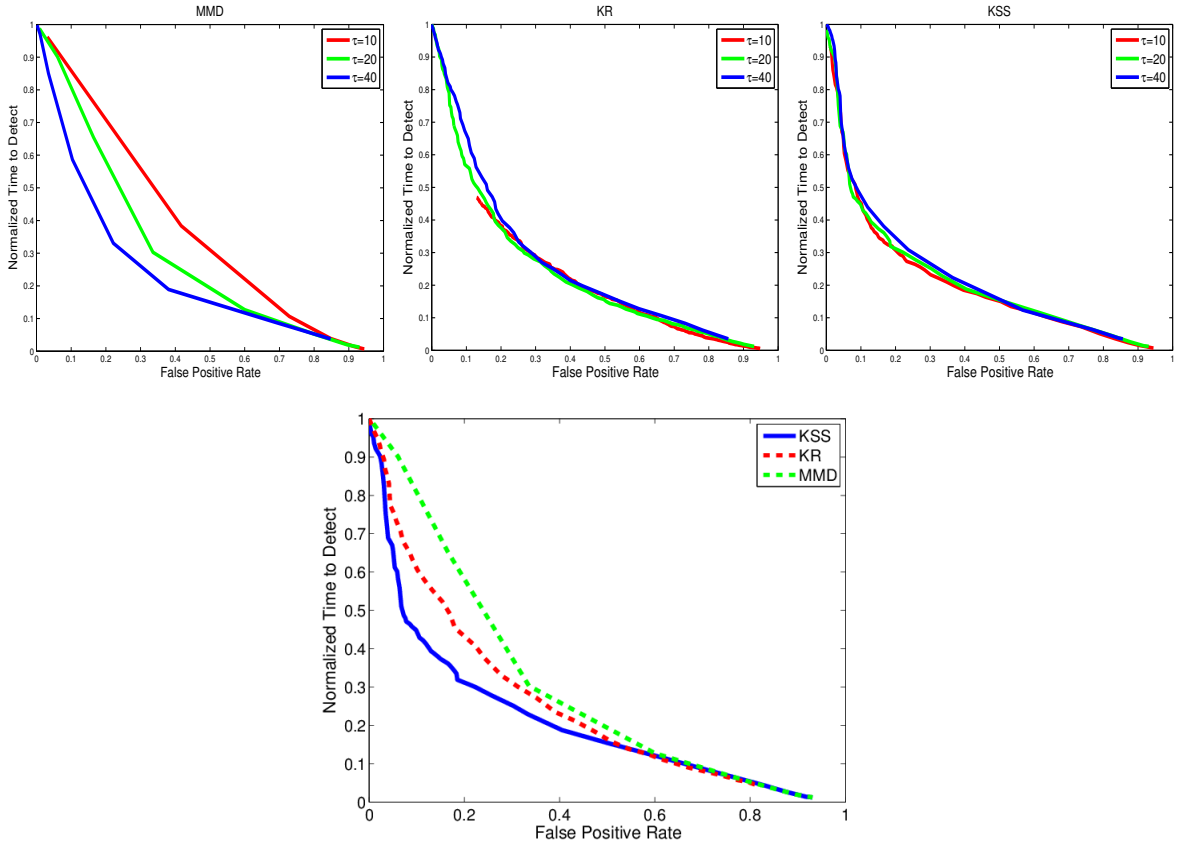

Figure 5.3: AMOC curves for the HAUS-PI dataset.

comparison graph, KSS curve is more close to ideal AMOC curve depicting its efficiency when compared to other two models supporting the theory.

5.4 Time Localization Accuracy

The measure of accuracy of detector to localize the event of interest is defined as Time Localization Accuracy. It can be determined in two ways - Rand Index(RI) and F1-Score.

5.4.1 RI

The Rand index or Rand measure in statistics, and in particular in data clustering, is a measure of the similarity between two data clusterings. A form of the Rand index may be defined that is adjusted for the chance grouping of elements, this is the adjusted Rand index. From a mathematical standpoint, Rand index is related to the accuracy, but is applicable

RI	KSS	KR	MMD
HAUS	0.72	0.71	0.70
UT	0.72	0.69	0.68

Table 5.1: Rand index

even when class labels are not used.

Definition Given a set of n elements $S = \{o_1, \ldots, o_n\}$ and two partitions of S to compare, $X = \{X_1, \ldots, X_r\}$, a partition of S into r subsets, and $Y = \{Y_1, \ldots, Y_s\}$, a partition of S into s subsets, define the following:

$$R = \frac{a+b}{a+b+c+d} = \frac{a+b}{\binom{n}{2}} \tag{5.2}$$

where,

a - #pairs of elements in S that are in the same set in X and in the same set in Y

b - #pairs of elements in S that are in different sets in X and in different sets in Y

c - #pairs of elements in S that are in the same set in X and in different sets in Y

d - #pairs of elements in S that are in different sets in X and in the same set Y

$a + b$ can be considered as the number of agreements between X and Y and $c + d$ as the number of disagreements between X and Y. This is a measure of the similarity between two data clustering. RI has been computed for the interaction segmentation against the ground-truth labels. A higher RI means better interaction localization. From the table, we can see that for the data sets experimented the localization has been improved from MMD to KR and from KR to KSS. In our HAUS dataset the detector is allowed to have more time to localize, i.e. we have more time before and after interaction in this dataset which helps for better localization, whereas the UTI sequences are short keeping concise to the interaction frames. KSS models seems to have good localization for both the datsets when compared to other which though did not vary much.

5.4.2 F1-score Curve

It is the efficiency measure of how well the detector can localize the event of interest. For a given time series t, if the detector output segment is y while the ground-truth truncated

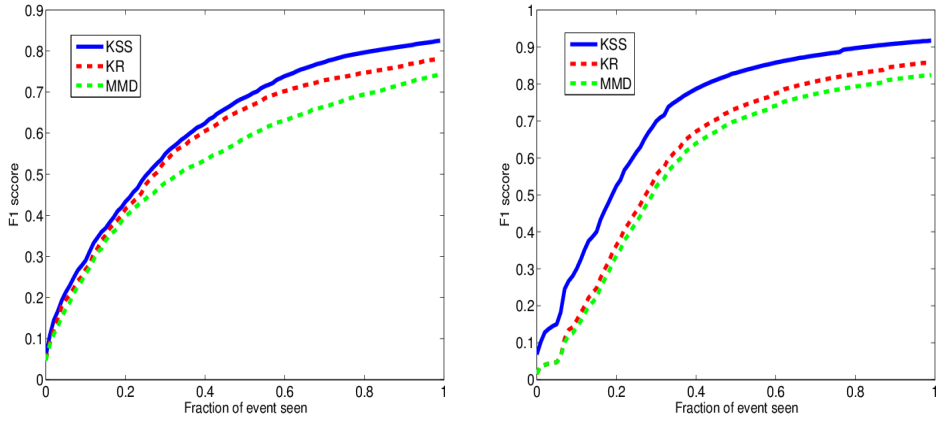

Figure 5.4: F-1 Score curves

segment is y*, the F1-score is given by

$$F1 := 2\frac{Precision * Recall}{Precision + Recall} \tag{5.3}$$

where, $Precision := \frac{|y \cap y^*|}{|y|}$ and $Recall := \frac{|y \cap y^*|}{|y^*|}$

F1 score for HAUS-PI (left), and UT-Interaction (right) datasets. Larger values of the F1 score for a given fraction of the interaction indicate better localization of ongoing interaction. We can draw similar conclusions from F1 score curve as we did from Rand Index Table, since these curves are just the graphical representation of time localization accuracy of the detector as RI table is numerically representation the same.

5.5 Recognition Accuracy

Recognition Accuracy is calculated depending on the system ability to classify the type of interaction in the video segments. Here we have considered nearly 50 sequences per class in HAUS and 10 samples per class in UT.

5.5.1 Recognition Accuracy Table

Below is the table containing the recognition accuracy of procedure for the temporally segmented sequences from all the three models compared against the ground truth annotation and for the two datasets - HAUS and UT (set1, set2). The results show the complexity of the new dataset collected and the efficiency of the framework for the easier data.

RA	KSS	KR	MMD	GT
HAUS	54.15	53.75	50.87	54.09
UT (set1)	0	0	0	95.08
UT (set2)	0	0	0	89.39

Table 5.2: Recognition Accuracy

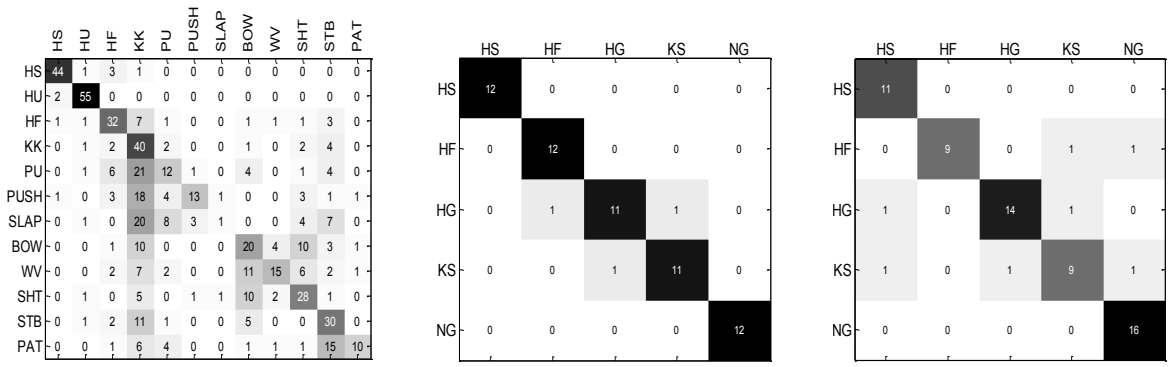

Figure 5.5: HAUS-PI Confusion Matrix

5.5.2 Confusion Matrices

In the field of machine learning, a confusion matrix, is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning one (here, multiclass libSVM). Each column of the matrix represents the instances in a predicted class, while each row represents the instances in an actual class. The name stems from the fact that it makes it easy to see if the system is confusing among the classes (i.e. commonly mislabeling or misclassifying one as another). Above are the confusion matrices for the HAUS-PI dataset(right) and UT-I dataset(left).

5.6 Parameter Search

Experiments have been done on the parameters like number of histogram bins of optical flow, the order of the system for HOOF features. The parameter range we tested on is 5 to 25 histogram bins with an increment of 5 bins and we have noted that the results are good with bin size 10. Whereas, we set the order range from 10 to 20 with an increment of 2 steps and observed that the order 15 showed better performance. We also experimented with frequency

i.e. with the number of frames for which we track the motion change, threshold to decide upon the motion, and motion histogram bins for MH features. 3-10 with an increment step of 1 is the range for frequency and threshold, among which 5 and 3 showed better results respectively. For the same search range of 5 to 25 of motion histogram bins 5 was good with order 10 of the system.

In our previous experiments [10], on two publicly available datasets TVHI and UTI generally used for this human-human interaction recognition purpose, experimentation on these parameters have been done on few different pairwise kernels like Geodesic, Tensor Learning kernel (TL), Tensor Product kernel, Direct Sum, Geodesic RBF kernels with and without distance as proximity. From those experiments as we found TL with distance proximity is the best kernel so we have only used that kernel.

5.7 Assumptions and Failures

All the above experiments hold good when we have the ground plane calibrations of the experimental area and also under the assumption that the tracking information available is accurate. This approach might not show better results in a cluttered scenes where there are more non interacting persons as they cause disturbance to the features. Also the model performance might not be as expected for the interactions not involving much body motions like talking and starring. Increasing more number of features like body joints to the HOOF and MH might make the approach more complicated deteriorating the results, this can be concluded from MOCAP dataset results implementing MMD [13].

Chapter 6

Conclusion

6.1 Summary

The purpose of performing temporal segmentation and recognition of human interactions addressed by a theoretically grounded approach that combines the geometry of RKHS with linear models has been introduced. The models, KR and KSS are extensively implemented and thoroughly evaluated against MMD approach on a old and a newly collected challenging datasets. Indicated by results we can draw a conclusion that the proposed and implemented framework is very promising, and can be an important part of a system for the analysis in real-time of human behavior from video.

Segmentation

The multidimensional data of non Euclidean space $\mathcal{S}$ is modeled effectively in the RKHS $\mathcal{H}$ through the KR, or the KSS model, by exploiting the kernel parity Hilbert space. This is achieved using the sequences of features describing activities.

In this thesis, the segmenting and recognizing has been done on the binary human interactions (interaction happening between person a and person b in the video). A sequence representation are obtained by tracking person i and j, and by aggregating their distance d_t, together with histograms i_t and j_t, describing the body motion of person i and person j, respectively, so that $y_t \doteq [i_t, j_t, d_t]^\top$. Pairwise kernels in particular *tensor learning kernel* is used, referring that the interaction between (i, j) and (j, i) is the same. The tensor

learning pairwise kernel used hence account for the symmetry as well as for the geometric structure of the input space $\mathcal{S}$-non Euclidean space. The kernel parameter γ is estimated with cross-validation.

Recognition

Human interactions are characterized by a temporally correlated sequence, assumed to be modeled by a KSS model. Geometric distances, algebraic kernels and information theoretic metrics can be used to compare LDSs, since KSS degenerates to KSS and S to Euclidean for the linear case. But for us, S is non-Euclidean space, we can use Binet-Cauchy kernels. As we used the same features - optical flow histogram bins and motion vector histogram bins along with the distance metric between the two persons interacting, we apply the Binet-Cauchy kernel that they refer to as kNLDS and which embeds kernel. Gaussian kernel based on the derived kernel distance has also been implemented and found to be more effective. To train a multiclass SVM classifier we use libSVM, with the above mentioned kernels.

6.2 Future Research

This work on modeling binary interactions could be improvised by including more complex features like gaze direction, as the interaction persons are assumed to face eachother. The video sequences that we used were all fully calibrated relative to a known ground plane. Using calibrated videos allowed the locations of peoples feet on the ground plane to be estimated from their head locations by assuming an average human height of 1.7 metres. These calibrations can also be used to approximate head size which might simplify obtaining gaze direction. Recently, the rapid development of depth sensors (e.g. Microsoft Kinect) provides adequate accuracy of real-time full-body tracking with low cost. This leaves us space to explore the feasibility of skeleton based features for activity recognition. Using Kinect sensor for extracting 3D skeleton joint features, which are known to best 3D features that outperformed other features for real-time interaction detection, might be a possible direction for future work. Replacing MILBoost (Multiple Instance Learning) classifier might improve interaction classification accuracy [52] if there exist irrelevant actions in the training data

[Irrelevant actions mean the frames around the peak of the interaction of interest. For example, the part of approaching, departing, and stretch arm in the hugging sequence is not actual hugging action, instead they are irrelevant actions or sub-actions].

Extending the dataset to include additional interaction categories or multiple viewpoints explore better human interaction representations.In the case of cluttered scenes rather than manual annotation which is effective but time consuming, using KLT Human Tracker might help making the framework applicable for real-time applications [KLT is known for robust object tracking in noisy environments].

Appendix A

Hilbert Space

A Hilbert space is an abstract vector space possessing the structure of an inner product that allows length and angle to be measured. It generalizes the notion of Euclidean space. It extends the methods of vector algebra and calculus from the two-dimensional Euclidean plane and three-dimensional space to spaces with any finite or infinite number of dimensions. An element of a Hilbert space can be uniquely specified by its coordinates with respect to a set of coordinate axes (an orthonormal basis), in analogy with Cartesian coordinates in the plane. When that set of axes is countably infinite, this means that the Hilbert space can also usefully be thought of in terms of infinite sequences that are square-summable.

A Hilbert space H is a real or complex inner product space that is also a complete metric space with respect to the distance function induced by the inner product. To say that H is a complex inner product space means that H is a complex vector space on which there is an inner product $\langle x, y\rangle$ associating a complex number to each pair of elements x,y of H that satisfies the following properties:

The inner product of a pair of elements is equal to the complex conjugate of the inner product of the swapped elements: $\langle y, x\rangle = \overline{\langle x, y\rangle}$.

The inner product is linear in its first argument. For all complex numbers a and b, $\langle ax_1 + bx_2, y\rangle = a\langle x_1, y\rangle + b\langle x_2, y\rangle$.

The inner product of an element with itself is positive definite: $\langle x, x\rangle \geq 0$ where the case of equality holds precisely when x = 0.

References

[1] J.C. Niebles, C.W. Chen, and L. Fei-Fei, "Modeling temporal structure of decomposable motion segments for activity classification," *In: Proceedings of the 12th ECCV, Crete, Greece*, 2010.

[2] J. Chen and A. Gupta, "Parametric statistical change-point analysis," *Birkhauser*, 2000.

[3] R.P. Adams and D.J. MacKay, "Bayesian online changepoint detection," *In: University of Cambridge Technical Report*, 2007.

[4] F.D. la, Torre, J. Campoy, Z. Ambadar, and J.F. Conn, "Temporal segmentation of facial behavior," *In: Proc. ICCV*, 2007.

[5] B. Schölkopf and A. Smola, *Learning with kernels: SVM, regularization, optimization, and beyond*, The MIT press, 2002.

[6] L. Ljung, *System identification: theory for the user*, Prentice-Hall, Inc., 2nd edition, 1999.

[7] E. Y. Chow and A. S. Willsky, "Analytical redundancy and the design of robust failure detection systems," *IEEE Transactions on Automatic Control*, vol. 29, no. 7, pp. 603–614, 1984.

[8] A. B. Chan and N. Vasconcelos, "Classifying video with kernel dynamic textures," in *CVPR*, 2007, pp. 1–6.

[9] R. Chaudhry, A. Ravichandran, G. Hager, and R. Vidal, "Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions," in *CVPR*, 2009, pp. 1932–1939.

[10] S. Motiian, K. Feng, H. Bharthavarapu, S. Sharlemin, and G. Doretto, "Pairwise kernels for human interaction recognition," in *Advances in Visual Computing*, 2013, vol. 8034, pp. 210–221.

[11] L. Hoegaerts, L. De Lathauwer, I. Goethals, J.A.K. Suykens, J. Vandewalle, and B. De Moor, "Efficiently updating and tracking the dominant kernel principal components," *Neural Networks*, vol. 20, no. 2, pp. 220 – 229, 2007.

[12] P. Honeine, "Online kernel principal component analysis: A reduced-order model," *IEEE TPAMI*, vol. 34, no. 9, pp. 1814–1826, 2012.

[13] D. Gong, G. Medioni, S. Zhu, and X. Zhao, "Kernelized temporal cut for online temporal segmentation and recognition," in *ECCV*, 2012, pp. 229–243.

[14] M. Hoai and F. De la Torre, "Max-margin early event detectors," in *CVPR*, 2012, pp. 2863–2870.

[15] Xuan X. and Murphy K., "Modeling changing dependency structure in multivariate time series," *In: Proc. ICML*, 2007.

[16] Y. Saatci, R. Turner, and C. Rasmussen, "Gaussian process change point models," *In: Proc. ICML*, 2010.

[17] F. Desobry, M. Davy, and C. Doncarli, "An online kernel chnage detection algorithm," *IEEE Trans. on Signal Processing*, vol. 53, 2005.

[18] Z. Harchaoui, F. Bach, and E. Moulines, "Kernel change-point analysis," *In: NIPS 21*, 2009.

[19] A.Y. Ng, Jordan M.I., and Y. Weiss, "On spectral clustering: Analysis and an algorithm," *In: NIPS Volume 14*, 2002.

[20] U. von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, , no. 17, 2007.

[21] F. Zhou, F.D. la Torre, and J.K. Hodgins, "Hieriarchial alined cluster analysis for temporal clustering of human motion," *Under review at IEEE PAMI*, 2011.

[22] E. Fox, Sudderth E., M. Jordan, and Willsky A., "Non-parametric bayesian learning of switching linear dynamical systems," *In: NIPS 21*, 2009.

[23] M. E. Brand and V. Kettnaker, "Discovery and segmentation of activities in video," *IEEE Trans. on Pattern Analysis and Machine Intelligence (TAPMI)*, 2000.

[24] J. Barbič, A. Safonova, J.Y. Pan, C. Faloutsos, J. K. Hodgins, and N. S. Pollard, "Segmenting motion capture data into distinct behaviors," in *Proceedings of Graphics Interface*, 2004, pp. 185–194.

[25] L. Zelnik-Manor and M. Irani, "Event-based video analysis," *In Proc. of IEEE CVPR*, vol. 2, pp. 123–130, 2001.

[26] N. Peyrard and P. Bouthemy, "Content-based video segmentation using statistical motion models," *In Proc. of British Machine Vision Conf. BMVC'02, Cardiff*, vol. 2, pp. 527–536, 2002.

[27] H. Zhong, J. Shi, and M. Visontai, "Detecting unusual activity in video," *In: Proc. CVPR.*, 2004.

[28] L. Zelnik-Manor and M. Irani, "Statistical analysis of dynamic actions," *IEEE PAMI*, vol. 8, 2006.

[29] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, "Automatic subspace clustering of high-dimensional data for data mining applications," *In Proc. of ACM SIGMOD*, 1998.

[30] J. Barbic, A. Safonova, J.Y. Pan, C. Faloutsos, J.k. Hodgins, and N.S. Pollard, "Segmenting motion capture data into distinct behaviors," *In: Proc. Grapics Interface*, 2004.

[31] P. Fearnhead, "Exact and efficient bayesian inference for multiple changepoint problems," *Stat. Comput.*, vol. 16, no. 2, pp. 203–213, 2006.

[32] Z. Harchaoui and O. Cappe, "Retrospective multiple change-point estimation with kernels," *in Proc. IEEE Workshop Statistical Signal Processing (SSP)*, pp. 768–772, 2007.

[33] J. Chen and A.K. Gupta, "Parametric statistical change-point analysis," *Cambridge, MA: Birkhauser*, 2000.

[34] J. K. Aggarwal and M. S. Ryoo, "Human activity analysis: A review," *ACM Computing Surveys*, vol. 43, no. 2, 2011.

[35] 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.

[36] 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*, Washington, DC, USA, 2004, CVPRW, IEEE Computer Society.

[37] 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.

[38] A. Patron-Perez, M. Marszalek, I. Reid, and A. Zisserman, "Structured learning of human interactions in TV shows," *IEEE TPAMI*, vol. 34, no. 12, pp. 2441–2453, 2012.

[39] M. Blaschko and C. Lampert, "Learning to localize objects with structured output regression," in *European Conference on Computer Vision*, 2008.

[40] C. Desai, D. Ramanan, and C. Fowlkes, "Discreminative models for multi-class object layout," in *International Conference on Computer Vision*, 2009.

[41] Y. Wang and G. Mori, "A discriminative latent model of object classes and attributes," in *European Conference on Computer Vision*, 2010.

[42] 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.

[43] L. Heoegaerts, L. De Lathauwer, I. Goethals, and B. De Moor, "Efficiently updating and tracking the dominant kernel principal components," in *Neural Networks 20*, 2007.

[44] Seon Joo Kim, Gianfranco Doretto, Jens Rittscher, Peter Tu, Nils Krahnstoever, and Marc Pollefeys, "A model change detection approach to dynamic scene modeling," in *AVSS'09*, 2009.

[45] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. Smola, "A kernel method for the two sample problem," in *NIPS*, 2007, pp. 513–520.

[46] I. Steinwart, "On the influence of the kernel on the consistency of support vector machines," *J. Mach. Learn Res.*, pp. 2:67–93, 2002.

[47] A. Gretton, K. Borgwardt, M. Rasch, B. Scholkopf, and A. Smola, "A kernel method for the two sample problem," *Technical Report 157,MPI for Biological Cybernetics*, 2007.

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

[49] S.V.N. Vishwanathan, A.J. Smola, and R. Vidal, "Binet-cauchy kernels on dynamical systems and its application to the analysis of dynamic scenes," *IJCV*, vol. 73, no. 1, pp. 95–119, 2007.

[50] W. Li and N. Vasconcelos, "Recognizing activities by attribute dynamics," in *NIPS*, 2012.

[51] Chih-Chung Chang and Chih-Jen Lin, "LIBSVM: A library for support vector machines," *ACM Trans. on IST*, vol. 2, pp. 27:1–27:27, 2011.

[52] Kiwon Yun, Jean Honorio, Debaleena Chattopadhyay, Tamara L. Berg, and Dimitris Samaras, "Two-person interaction detection using body-pose features and multiple instance learning," *CVPR Workshops*, 2012.