MODELING SCENES AND HUMAN ACTIVITIES IN VIDEOS

by

ARSLAN BASHARAT
B.S. Ghulam Ishaq Khan Institute, Pakistan
M.S. University of Central Florida

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the School of Electrical Engineering and Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Fall Term
2009

Major Professor: Mubarak Shah

ABSTRACT

In this dissertation, we address the problem of understanding human activities in videos by developing a two-pronged approach: *coarse* level modeling of scene activities and *fine* level modeling of individual activities. At the coarse level, where the resolution of the video is low, we rely on person tracks. At the fine level, richer features are available to identify different parts of the human body, therefore we rely on the body joint tracks. There are three main goals of this dissertation: identifying unusual activities at the coarse level, recognizing different activities at the fine level, and predicting the behavior in order to synthesize activities at the fine level. The summary of the three proposed solutions is presented in the following.

The first goal is addressed by modeling activities at the coarse level through two novel and complementing approaches. For this purpose, we rely on the tracks of all the moving objects in the scene observed by a static camera. First approach learns the behavior of individuals by modeling the patterns of motion and size of objects in a compact model. The proposed method provides a higher-level process to the traditional real-time surveillance pipeline for identifying unusual activities and feeding back the learned scene model to improve object detection. Pixel level probability density functions (pdfs) of appearance have been used for background modeling in the past, however modeling pixel level pdfs of object speed and size from the tracks is novel. Each pdf is modeled as a multivariate Gaussian Mixture Model (GMM) of the motion (destination

location & transition time) and the size (width & height) parameters of the objects at that location. Output of the tracking module is used to perform unsupervised EM-based learning of a GMM at every pixel location. Second approach learns the interaction of object pairs concurrently present in the scene. This can be useful in detecting more complicated activities that the first approach cannot model. We use a higher dimensional Kernel Density Estimation (KDE) model in order to create this model. Mean shift is used for sample refinement followed by Markov Chain during testing stage. The proposed model is successfully used to detect abnormal activities like illegal jaywalking, person drop-off and pickup, etc. Most object path modeling approaches first cluster the tracks into major paths in the scene, which can be a source of error. We avoid this by building local pdfs that capture a variety of tracks which are passing through them. We also show the improvements in object detection through the feedback of the learned scene model.

The second and third goals of modeling human activities at the fine level are addressed by employing non-linear dynamical systems. We show that such a model can be useful in recognition and prediction of the underlying dynamics of human activities. In the case of human activities, we use the trajectories of human body joints as the time series data generated by the underlying dynamical system. For this work we have borrowed the relevant key concepts from chaos theory and developed methods to utilize them to solve the problems at hand. Next, we explain the proposed recognition and synthesis methodologies based on the chaotic modeling of human activities.

We introduce a recognition framework that uses concepts from the theory of chaotic systems to model nonlinear dynamics of human activities. The observed time series data is used to reconstruct a phase space of appropriate dimension by employing a delay-embedding scheme. The properties

of the reconstructed phase space are captured in terms of dynamical and metric invariants, which include the Lyapunov exponent, correlation integral, and correlation dimension. The underlying dynamical system is eventually represented by a composite feature vector containing these invariants. Our contributions in this work include: investigation of the appropriateness of the theory of chaotic systems for human activity modeling and recognition, a new set of features to characterize nonlinear dynamics of human activities, and experimental validation of the feasibility and potential merits of carrying out activity recognition using methods from the theory of chaotic systems.

Finally, we also propose a framework for predicting the time series data observed in human activities. We utilize concepts from chaos theory in order to predict the behavior of a nonlinear dynamical system which exhibits deterministic behavior. Observed time series from such a system can be embedded into a higher dimensional phase space without the knowledge of an exact model of the underlying dynamics. Given an initial condition, the predictions in the phase space are computed through kernel regression. This approach has the advantage of modeling dynamics without making any assumptions about the exact form (linear, polynomial, radial basis, etc.) of the mapping function. The predicted points are then warped back to the time series format. We demonstrate the utility of these predictions for human activity synthesis and tracking. Our main contributions are: multivariate phase space reconstruction for human activities, a deterministic approach in contrast to the popular noise-driven approaches, and activity prediction through kernel regression in the phase space.

To my loving parents and beloved wife

ACKNOWLEDGMENTS

I am very grateful to Dr. Mubarak Shah for his continued support and guidance over the past several years. His persistence and support was key to the successful completion of this work. The research environment he provided was greatly beneficial in provoking analytical thinking and nurturing ideas. I would like thank the PhD committee members, Dr. Niels Lobo, Dr. Annie Wu and Dr. Xin Li, for their time and valuable comments. I would also like to thank my wife, Jamila, and my parents for their love and constant encouragement. I would also like to thank my colleagues, many of which I have collaborated with over the years, including Yun Zhai, Alexei Gritai, Yaser Sheikh, Omar Javed, Asaad Hakeem, Saad Ali, Saad Khan, Kairoek Choeychuen, Kittiya Khongkraphan and several others.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

1.1 Background and Motivation

The understanding of human activities in videos has attracted the attention of many in the computer vision research community. This technology can be useful in a variety of applications including, but not limited to, security & surveillance, human computer interaction, robotics, and multimedia. All of these application domains will have a significant impact on various aspects of our everyday lives. Security & surveillance systems can be important for the public safety at airports, train stations, and large parking lots. The safety of various resources at warehouses, power houses, military installations, etc. is also of significant interest to security agencies. In the case of human computer interaction and robotics, a key objective is to automatically recognize different gestures to which the machine then responds to appropriately. For instance, in the recent years there has been an increased interest in developing camera equipped gaming consoles where the goal is to create a more realistic interactive experience. In the case of multimedia and information retrieval, there is significant interest in retrieving videos containing specific types of activities (e.g. dancing, fighting, kissing, etc.) from large databases of movies and broadcast television videos. This could prove to be greatly beneficial for organizations and individuals with rapidly growing video archives.

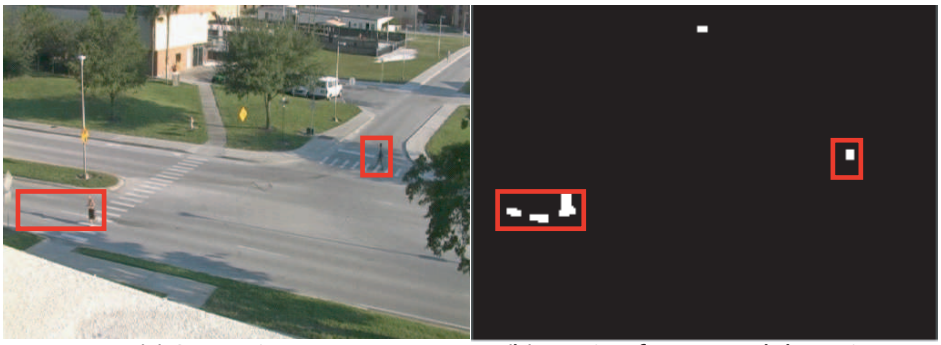
In order to build a robust system, which identifies various types of human activities, one should consider several factors that can affect the system design choices. In the following we discuss some

of these important factors and explain how they affect the choices we have made in the proposed approach.

1. **Application Domain:** The activities of interest and the significance of the fine details could vary depending on the application domain. For instance, in the case of a surveillance system, the primary interest is typically in the identification of unusual behavior (e.g. falling down, jaywalking, jumping over a fence, etc.). On the other hand, in the case of human computer interaction, the primary interest is typically in the details of specific activities (e.g. waving with one hand, waving with two hands, kicking with right foot, etc.). Therefore, activity models should be devised in order to accommodate the application domain of interest.

2. **Video Quality:** The quality of video is an important factor that should be considered while devising a robust activity recognition system. The quality could depend on the resolution, color contrast, frame rate, etc. of the source video. For instance, in higher resolution scenes, it is possible to extract useful motion signatures of the individual parts of a human body. The same features would be far less useful when the resolution is very low, like in the case of video from a far-view surveillance camera. Figure 1.1 shows one such example where the foreground detections capture only a part of the bodies of the two individuals. Hence, the quality of the source video should be considered when selecting the type of observed feature (e.g. body joint trajectories, person trajectory, body shape representation, etc.).

3. **Learning Paradigm:** A learning based approach can be used to recognize different human activities. It has the advantage of robustness to intra-class variations. The learning can be



(a) Source image (b) Moving foreground detections

Figure 1.1: The quality of moving foreground detections at the coarse level is low. Typically the image of human body is comprised of only a few hundred pixels. Notice that only a part of the human body is detected as the foreground.

supervised or unsupervised depending on the type of training data available. In the case of supervised learning, we can learn a separate model for each type of activity and possibly assign a semantic label to the class. On the other hand, in case of unsupervised learning, we do not have the luxury of separating different activity classes and thus can only devise an "outlier detector". This can be useful in the case of surveillance system when the goal is to detect unusual activities in the scene.

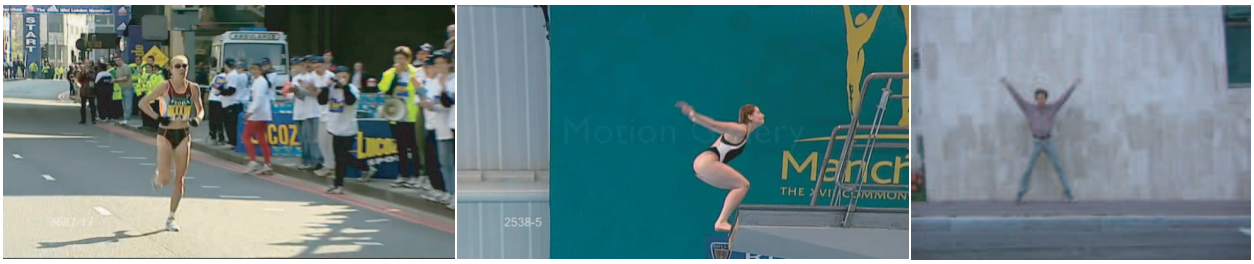
Considering the aforementioned factors, we approach the problem of recognizing human activities by adopting two different approaches for modeling activities at the *coarse* and the *fine* level. The coarse level comprises of activities defined by the global motion of the object (person trajectory), low resolution, and unsupervised learning. The fine level comprises of activities defined by the local motion of body parts (joint trajectories), higher resolution, and supervised learning. Figures 1.2 and 1.3 present samples of activities at the coarse and fine levels respectively. In this

(a) Walking on the sidewalk (b) Sitting on the sidewalk (c) Unusual path

Figure 1.2: Sample activities at the coarse level, typically observed from a stationary surveillance camera.

dissertation, we contend that a *one size fits all* approach is not most appropriate and we should devise two different methods for the two levels.

We first show that we can model an activity well enough to recognize it, in addition, we can also use the model to recreate or *synthesize* the activity. This has several uses for the underlying task of understanding human activities. It can be useful for qualitative and quantitative validation of the learned model being used for recognition. The activity synthesis has wide range of applications in the area of computer graphics and animation. Predictions are computed in order to synthesize an activity. These predictions can also prove to be vital for the task of accurately and efficiently localizing and tracking different parts of the parts of the human body. We also demonstrate the results of human body parts tracking on a set of periodic action from a standard action dataset. Next, we highlight the main goals of this research along with the proposed solutions for each one of these goals. In Section 1.3, we present the descriptions of the proposed solutions and the summary of our contributions.



(a) Running (b) Diving (c) Jumping Jack

Figure 1.3: Sample activities at the fine level. Typically a human body is covered by a few thousand pixels and the individual body parts have a few tens or even hundreds of pixels.

1.2 Goals

The goals of this work are to detect abnormal activities at the coarse level and to recognize and synthesize activities at the fine level. We present following three solutions to address these goals.

1. Statistical scene modeling for unusual activity detection.

2. Chaotic invariants for human activity recognition.

3. Chaotic modeling for human activity prediction.

1.3 Outline of This Research

In this section we introduce the two different approaches for modeling coarse and fine level activities. Figure 1.2 shows samples of a few activities that are being modeled at the coarse level. In this case a scene model will be learned which is composed of all the observed activities. On the other hand, Figure 1.3 presents a few examples of the activities that can be modeled at the fine

level. This involves modeling of individuals instead of the whole scene by capturing the details of the body part motion.

1.3.1 Coarse Level Activity Modeling

We approach this problem by learning, in an unsupervised manner, all the activities in the scene. Once the *usual* activities have been learned, the goal is to identify any *unusual* activities in the scene. This kind of approach is particularly useful for the security and surveillance application domain. Such a model can learn the patterns of various types of activities that could otherwise be too abstract to be distinguished by separate classes. For instance, such an approach is able to distinguish between a person jaywalking in the middle of the road and a person using a crosswalk, although both of them are walking. This proves to be useful in identifying semantically meaningful activities because they are unusual considering what system has learned automatically. Other interesting scenarios that motivated us included automatic detection of: one-way traffic violations, speeding, illegal u-turns, collapsed individual on the sidewalk, restricted area violation, etc.

In order to learn patterns of object motion in a scene we propose two novel and complementing models based on statistical learning. The first model is useful for learning behaviors of individual objects only, while the second one has the benefit of learning the relationship of objects in pairs. The goal is to learn a distribution that presents typical behavior during the training phase and can be used to identify abnormal activities during the testing phase. We use a local GMM based pdf at every pixel in the first model, and a global KDE based pdf for the whole scene in the second model.

The models presented here have following novel contributions:

- We propose a new and intuitive approach to model object parameters (motion and size) by using a pdf at every pixel location. Stauffer and Grimson's [99] approach has been used for modeling appearance for several years, but the proposed model of motion and size at pixel-level is novel.

- In addition, we present a second novel model that captures the relative relationship of objects in pairs. Saleemi *et al.* [95] have presented a similar single object model recently, but the a model for object pairs is novel.

- Unlike most of the previous approaches, our models do not require extraction of major paths in the scene and is learnt directly from the individual tracking observations.

- The motion parameters are used to capture the *local* velocity of an object, as well as the *global* velocity through the track. This helps in detecting the anomalous motion patterns that cannot be captured by local analysis only.

- The presented models can be used to perform online learning of the evolving motion patterns in the scene.

- We utilize this model to provide pixel-level parameter feedback to the background subtraction module in order to improve object detection. Instead of constraining the object detection module by having fixed parameter values throughout the scene, we present a method to provide different pixel-level parameter values using the learnt scene model. Two parameters:

Minimum size of the foreground objects and the background learning rate, have been used to improve object detection by our approach.

1.3.2 Fine Level Activity Modeling

Our model for fine level is based on non-linear dynamical systems. We use key concepts from chaos theory which enable us to create models of dynamics without actually having a parametric form of the dynamical system. This is particularly useful when working with the experimental data and the underlying model of the dynamical system is unknown.

Input to a dynamical system is a sequence of time series observations. In our model, the time series data is received from trajectories of human body joints. Observed time series, in this case, can be embedded into a higher dimensional phase space without the knowledge of an exact model of the underlying dynamics. This embedding warps the observed data into a *strange attractor* in the phase space, which provides precise information about the dynamics involved. After the embedding, one can extract invariant features for recognition or perform regression for prediction.

1.3.2.1 Chaotic Invariants for Human Activity Recognition

The representative shape of the strange attractor is captured through a set of features that will be useful to identify the underlying dynamics uniquely. The properties of the reconstructed phase space are captured in terms of dynamical and metric invariants which include the Lyapunov exponent, correlation, and correlation dimension. We use a composite feature vector of invariants for classification.

Our contributions in this work include:

- Investigation of the appropriateness of the theory of chaotic systems for human activity modeling and recognition.
- A non-linear dynamical system based representation of an action that without assumptions about the mathematical form. Previous models have assumed a linear model or assumed a linear combination of non-linear basis functions.
- A new set of features to characterize nonlinear dynamics of human activities.
- Experimental validation of the feasibility and potential merits of carrying out activity recognition using methods from the theory of chaotic systems.

1.3.2.2 Chaotic Modeling for Human Activity Prediction

Once the training time series data has been embedded into phase space, we extract the information about the underlying from the strange attractor and utilize it to predict future observations. Given an initial condition, the predictions in the phase space are computed through kernel regression. The predicted points are then warped back to the observed time series.

Our main contributions in this work include:

- Predicting dynamics without making any assumptions about the exact form (linear, polynomial, radial basis, etc.) of the mapping function.
- Multivariate phase space reconstruction for human activities.

- A deterministic approach to model dynamics in contrast to the popular noise-driven approaches.
- Video synthesis and action tracking from kernel regression in the phase space.

1.4 Organization of Dissertation

The rest of the dissertation is organized as follows. In Chapter 2 we cover relevant research in the related areas. We also explain how the proposed research contributes to the literature in perspective of the previous work. Chapter 3 presents our approach for learning object motion patterns in a stationary camera. We present results of anomaly detection and scene model feedback to improve object detection. We present two complementing models for learning object motion patterns of single objects, as well as object pairs. Chapter 4 presents the details of a novel chaos theory based approach for human activity recognition. We provide the details of representing actions through phase space embedding and chaotic invariants for recognition. Chapter 5 presents our approach for predicting dynamics through kernel regression in phase space also used for recognition. We also provide the multivariate extension of phase space embedding for better predictions. We show the application of these predictions for human action synthesis, human body parts tracking, and dynamic texture synthesis. We present experimental results of the three approaches using the published data sets. Finally, in Chapter 6 we conclude this dissertation with discussion and review of future directions.

CHAPTER 2: LITERATURE REVIEW

In this chapter we cover the most relevant work in the research literature. We cover the topics of scene modeling for anomaly detection, activity recognition, dynamical systems based models as used in activity modeling, and activity synthesis. We present the merits and demerits of many of these approaches while referring to other similar ones. We also explain the contributions of our work in the context of the previous work.

2.1 Scene Modeling for Abnormal Behavior Detection

Scene modeling, in this dissertation, refers to the modeling of normal object motion in the scene. Such a model is typically used to learn the typical behavior in the scene and differentiate this from any unusual object behavior. The term "scene modeling" is not used here in context of scene content matching in domain of video matching and retrieval [8].

Analysis and modeling of motion patterns for surveillance scenes has been studied by several researchers. Buxton [16] provided a detailed review of the models that have been used for learning scene activity. Johnson *et al.* [57] presented a vector quantization based approach for learning typical trajectories of pedestrians in the scene, but they require entry/exit points to be marked manually. Grimson *et al.* [47] used location, velocity and size to classify activities. The activities are classified using a B-tree based approach called Numeric Iterative Hierarchical Cluster method and

the co-occurrence statistics in the quantized feature space. In [89], Remagnino *et al.* use velocity and aspect ratio to classify different tracks into vehicle or person. They utilize a Bayesian classifier for this task and an HMM model to capture common events in the scene. Makris *et al.* [73] have presented a technique in which different regions of the scene are labelled as entry/exit zones, junctions, paths and stop zones. This model provides a set of scene attributes but lacks the object size-based anomaly detection. Saleemi *et al.* [95] proposed a single Kernel Density Estimate (KDE) model for the whole scene, which requires to save all training data. Their approach does not address anomalies due to object size and only focuses on the object velocity. In comparison, we present a more compact GMM based model when modeling the motion of single objects. We rely on KDE model only in the more complex case of object pair motion.

Hu *et al.* [53] present a recently published technique in which the tracks are spatially and temporally clustered into different motion patterns. Each of these motion patterns is divided into several segments; each segment is modeled by a Gaussian model of speed and size. Anomaly detection and path prediction are the two applications of this approach. Wang *et al.* [106] have presented another approach in which the tracks are clustered into vehicle and pedestrian paths. Their model provides the source/sink information along with capability of abnormality detection.

Morris *et al.*[79] have recently presented a catalogue of various approaches for trajectory clustering in the domain of scene modeling with the goal of activity detection. They cover six different distance measures that have been used before for this task. They experiment with six different clustering approaches including direct, divisive, agglomerative, hybrid, graph, and spectral. If one chooses to follow the route of clustering the trajectories, this can serve as a good starting point.

We present a model here that avoids the errors related to clustering of trajectories. Instead, we approach this problem by using a pdf model that can either represented by GMM at every pixel or one global pdf through KDE. The risk involved in the parameter selection of the former model is avoided by using an automatic EM approach by Figueiredo and Jain [39].

In the past year or two there has been an increased interest in detection of unusual activities in crowded situations[64, 75, 5, 65]. Kim *et al.*[64] proposed a space-time Markov Random Field (MRF) model for detecting abnormal activities in the scene. They learn the distribution of local optical flow using a mixture of probabilistic principal component analyzers. For testing the learnt model and MRF graph is used to compute a maximum a posteriori estimate. They create models at the local spatial neighborhood level. Mehran *et al.*[75] have presented an approach based on social force model with particle flow field to model the motion of individual in crowded environment. The model of the normal crowd behavior is extracted from the spatiotemporal volumes representing reasonable particle interaction. A bag of words representation is used for detecting abnormal behavior in comparison to the scene model. Ali and Shah [5] had initially utilized flow fields as advection of the optical flow computed at every frame and integrated through time. Their approach was based on Lagrangian particle dynamics for crowd flow segmentation. The Finite Time Lyapunov Exponent was used in order to determine coherence of particle dynamics through the flow. Our use of Lyapunov Exponent as a part of chaotic invariants for human activity recognition is relevant to this work. They have also shown the identification of new crowd segments as a way to perform abnormal behavior detection in crowds. Lastly, Kratz *et al.*[65] presents aimed at addressing crowded situation and the goal is to identify regions with unusual activity in the scene. They

use GMM based models of spatiotemporal gradient in video cubes. The cuboids are compared using the symmetric Kullback-Leibler divergence. Temporal relationship is finally captured through Hidden Markov Models. The decision about the normal vs. abnormal behavior is taken based on the likelihood of an observation sequence in an HMM, using the traditional forward-backward algorithm. A common theme in these, and other, crowd analysis approaches is that they do not rely on object tracking. The observed feature is typically optical flow or a derived flow field. The scene modeling approach presented in this dissertation would not be directly applicable to such crowded scenarios. However, the first proposed scene model based on GMM can be modified to learn the distribution of local optical flow instead of the tracking data. The new model would then closely follow the behavior of other statistical learning based approaches discussed above.

Scene modeling can also be used to feedback the scene knowledge into object detection module. In [49], Harville proposed an approach with positive and negative feedback to background subtraction for adjusting the learning rate and improving foreground detection. Tian *et al.* [111] detected the static regions that were wrongly modeled as the background. In addition to learning rate, there are other parameters that affect the background subtraction and could benefit from the feedback. In this approach we use the same scene model to provide feedback in order to update minimum object size and background learning rate parameters. The unique aspect of our approach is the use of the same scene model for both anomaly detection and improving object detection.

One common factor in most of the related work is the estimation of main motion paths in the scene. Techniques presented in [53, 59, 100, 106] use multiple features of observed tracks for clustering tracks into the main paths of the scene. We argue that the explicit estimation of

these paths is not necessary for typical applications of a scene model including anomaly detection and improving of object detection. In addition, these approaches only capture the instantaneous velocity, however in the proposed approach we integrate larger transition times. This captures the *global* properties of the track and therefore does not require the estimation of the main paths in the scene.

2.2 Human Activity Recognition

Human action/activity recognition is an important area of research in the field of computer vision. The pioneering research efforts [3, 50, 52, 68, 87, 92] in this area tried to address the problem in early eighties by modeling the articulated body skeleton for human activity analysis. For instance, Akita [3] compared the learned body model with the key-frames of the sequence to test the presence or absence of the activity, whereas Lee *et al.* [68] employed a 3D configurations of the model and tried to find the best matching with the 2D motion-based segmentation of the image. On the same lines, Hogg [52] studied the motion of a walking figure using an articulated model. Since then a huge body of literature that addresses different aspects of the activity recognition problem has been published. Comprehensive reviews of this research has been presented in a number of survey papers over the years [24, 2, 1, 44, 76, 17, 66]. Readers are referred to these survey papers for the in depth coverage of the field. In this section, we will limit ourselves to some of the most influential and relevant part of this literature.

In general, approaches for human activity analysis can be categorized on basis of the representation used by the researcher. Some leading representations are learned geometrical models

of human body parts, space-time pattern templates, appearance or region features, shape or form features, interest point based representation and motion/optical flow patterns. In early years representation based on appearance features was a popular approach. The general methodology was to learn the appearance model of human body or hand etc., and match it explicitly to a target video sequence for activity or gesture detection [14, 33, 98, 113, 42]. The temporal aspects of an activity were handled by either training hidden markov models (HMM) or its different variants. But soon it was realized that this representation is limited in its ability to handle realistic situations as it is prone to changes in the appearance of the actor. However, some recently published papers [78, 77, 56] are still pursuing appearance based representation for activity recognition in images by searching for static postures using the appearance of the whole human body or parts of the body. An important short coming of these approaches is the localization of body part which itself is a very hard problem. We believe that use of only appearance based information for activity recognition is counter intuitive as activities are a temporal or dynamic entity.

Popular shape based representations include edges [22] and silhouettes of human body [28]. The idea behind shape based representation is that an activity consists of a series of poses which are detectable from a single frame. Each pose can be encoded using the shape features and single frame recognition in turn can be extended to more than one frame for robust activity recognition. The silhouette based representation was recently extended to characterize actor's body outline through space and time [114, 11]. This is done by stacking the individual silhouettes detected in each frame giving rise to a three dimensional volume. Yilmaz *et al.* [114] used surface properties of this volume for activity recognition. While Moshe *et al.* [11] used solution of poisson equation

to extract space time features of the volumes. Note that these approaches can also be categorized under a volume based representation. Although these approaches have demonstrated robust performance on a number of activities, they lack the ability to incorporate the rich motion information in their representation as they concentrate on the properties of the surface of the volume. That is their emphasis is more on capturing the *form* of the human body.

The approaches based on volumetric analysis of video for activity recognition. Ke *et al.* [61] extended the two dimensional Haar features to three dimensions and learned a cascade of boosted classifiers. In [62], they later addressed action detection in cluttered scenes by using partial matching of action volumes. Shechtman *et al.* [97] employed a three dimensional correlation to match the actions in the space time volume. Mahmood *et al.* [102] also used volume representation for activity recognition. One benefit of the volume based approach is that there is no need to build complex models of body configuration and kinematics, and recognition can be done directly on the raw video. Another important direction of research that has gained much interest recently is the use of space time interest points and their trajectories for activity analysis. Work by Laptev *et al.* [67], Oikonomopoulousm *et al.* [82] and Dollar *et al.* [35] belongs to this category. The main strength of this representation is its robustness to occlusion as one does not need to track or detect the whole human body.

The features based on motion information and optical flow, which are more relevant to our current work, have been used by a number of researchers [37, 70, 58, 112]. For instance, Bobick *et al.* [12] introduced motion energy image (MEI) as way of describing cumulative spatial distribution of motion energy in the given sequence. This description of motion is then matched

against stored models of the known activities. The MEI descriptor was later augmented with motion history image (MHI) in [13], where each pixel intensity in MHI is described as a function of of motion energy. Recently, Weinland et al. [108] extended this representation to handle different viewpoints. In optical flow based approaches [70, 112, 10] the idea is to directly use the optical flow as a basis for deriving a representation that can be used for recognition. Little *et al.* [70] used spatial distribution of the magnitude of the optical flow for deriving model free features, while Ju *et al.* [58], Yacoob *et al.* [112] and Black *et al.* [10] proposed PCA based analysis of optical flow for facial motion and expression analysis.

Chaudhry *et al.*[26] have recently presented an approach where they model an action through nonlinear dynamical system (NLDS). The are using a histogram of oriented optical flow (HOOF) as the observation in each frame. The sequence of HOOFs is then used as the input time series of the NLDS. They use generalization of Binet-Cauchy kernels to NLDS in order to compare two HOOF time series. They claim to be the first ones to have used a complex descriptor, like HOOF, instead of a set of trajectories of human body joints or a series of pose descriptors containing the joint angles etc. This approach uses the kernel for projection to a higher non-Eucledian space in order to compute distance between to HOOF time series. Such a model is useful for the computation of action recognition but cannot be generalized to action representation for other tasks like prediction, tracking, etc. The model presented in this dissertation provides a strong representation in the phase space.

In addition, we would like to mention that a different paradigm for activity recognition has also been advocated over the years where 3D information of human postures and dynamics is analyzed

[34, 18]. The projection of these 3D models are used to test whether desired activity is present in the given frame or not. Due to explicit construction of 3D models, these approaches are able to handle view invariance but suffer from the difficulty of recovering 3D structure of the articulated objects.

Our present work is more related to the approaches of learning dynamical models over the state space that represent human motion ([9, 81, 15]). Specifically, the method by Bissacco *et al.* [9] used a parametric skeletal model of a moving person and learned a linear dynamical model, while Bregler [15] proposed a mixed-state statistical model with a finite state automaton at the highest level to switch between local linear models to cater for the nonlinear dynamics of human motion. Later on [86, 66] attempted to integrate the nonlinear dynamics directly into the model, rather than using an external mechanism to control the switching.

2.3 Dynamical Systems and Video Synthesis

Polana and Nelson [85] classified visual motion into three classes: motion events, activities, and temporal textures. Motion events (e.g. sitting, opening window) don't exhibit temporal or spatial periodicity. Activities (e.g. walking, jumping) are formed by the motion patterns that are periodic in time and localized in space. Temporal textures (e.g. waves on water surface, smoke) present statistical regularity but have indeterminate spatial and temporal extent. We focus on the temporal regularity of the last two classes. For this we rely on the powerful tools from chaos theory to model deterministic dynamical systems [60].

In computer vision, dynamical systems have been used in a variety of applications, including human motion (activity) modeling [9, 15, 40] and dynamic textures [25, 36, 45, 71, 115, 107], and tracking [109]. Most of these approaches model underlying system dynamics by using linear systems, while others use nonlinear dynamical systems. In many cases, nonlinear approaches provide a more accurate model but have to approximate the parametric form of the underlying system. This parameter learning may be imprecise and that can be a source of error. Our approach belongs to the category of the nonlinear dynamical systems that use nonparametric model, which therefore does not require parameter learning.

Many of the previous approaches for dynamical systems rely on stochastic noise-driven linear [36, 115] and nonlinear dynamical systems [25]. Instead, we show that the typical dynamic textures can be modeled accurately by deterministic dynamical systems. The detailed experimental validation proves our argument. In [69] and [71], authors present approaches for learning nonlinear manifold for the observed time series. We have compared our method with [71] and show that our approach generates more realistic dynamic textures, because it does not suffer from the errors due to imprecise learning.

Time series modeling and prediction has been an active area of research due to the wide variety of applications in the financial market, weather, biology, etc. The initial approaches typically relied on AR, MA, or ARMA univariate models. More sophisticated approaches rely on nonlinear modeling [23] and state space projection of the time series [86]. Our approach has both of these properties. Ralaivola *et al.* [86] present an approach for time series prediction based on kernel trick and support vector regression. In comparison, our approach is based on delay embedding [104]

and kernel regression [80]. Delay embedding generates the unique *strange attractor* that can be used for system modeling and classification. [60].

Wang *et al.* [40] have presented another strong model for human motion. They propose a non-parametric dynamical system based on Gaussian processes. This approach is only demonstrated for human motion and not for the higher dimensional data, such as dynamic textures. The case of dynamic textures is more challenging than human activity because of the higher dimensional observations and more irregular variations in the system state. Our approach is general enough to be applicable to both human activities and dynamic textures. In addition, our method does not require multiple exemplars for training in order to learn a particular activity, making it more practical.

Huang *et al.* [54] have recently presented a new approach of human action synthesis in 3D video using surface motion graphs. Their goal is to allow a user to specify a set of key poses needed in the output video. Their goal is to use the available poses and in the database and minimize the cost of transition between the key poses. They construct the novel 3D video by finding the optimal path in the surface motion graph among the key poses specified by the user along with location and timing. They use integer linear programming for finding the optimal set of poses. This type of framework is suitable for building composite activities (like walking and then running) that are based on different combinations of individual activities (like running, walking, etc.). The activity synthesis approach presented here focuses on the synthesis of individual activities instead of activity transitions.

A common theme of all these approaches is that they approximate the true motion dynamics by putting constraints on the type of the dynamical model. In addition, they require very detailed mathematical and statistical modeling which involves assumptions about the probability distribu-

tions of stochastic variables of the model, development of inference methods, and algorithms for learning parameters of the distribution using a large data set. To overcome some of these difficulties, we are proposing a framework that captures the true non-linear dynamics of the human motion, and generates a more richer set of features by directly working with the experimental data. In addition, our method is not a statistical learning method therefore does not require large training data, instead strong discriminative features can be derived just from one example activity.

2.4 Summary

We have presented an overview of the related research in the areas of anomaly behavior detection, activity recognition, dynamical systems and activity synthesis. We discussed pros and cons of various approaches in the literature. We also explained how the proposed work is aimed at filling the void in the literature. Our approach for anomalous activity detection is based on unsupervised learning, models motion of single objects as well as object pairs, avoids errors related to clustering tracks, and reuses the same scene model for improving object detection. We have presented a novel approach to model human activities as a dynamical system in the phase space. To the best of our knowledge, we have used the relevant concepts from chaos theory and non-linear dynamical systems for the first time to represent human activities and dynamic textures in computer vision literature. We have used a new set of features (chaotic invariants) for recognizing activities and proposed a new approach (kernel regression in phase space) for predicting human activities and dynamic textures.

In the next three chapters we present the details of our approaches for unusual activity detection, activity recognition, and activity prediction. The following chapter presents the details of our statistical learning approach to detect unusual activities at the coarse level in the scene.

CHAPTER 3: SCENE MODELING FOR UNUSUAL ACTIVITY DETECTION

3.1 Introduction

Automated video surveillance is crucial for the security of various sites including airports, train stations, military bases, and many other public facilities. There have been significant advances in automated visual surveillance systems in the recent years [30, 74]. A modern surveillance system is expected to not only perform basic object detection and tracking, but also to interpret object behaviors. This higher level interpretation can have several applications including abnormal behavior detection, analysis of traffic trends, and improving object detection and tracking. In this chapter, we focus on the problem of interpreting the output of the object detection and tracking module in order to gather knowledge about the scene. This knowledge is used to build a scene model which can be used to detect abnormal motion patterns and to enhance the surveillance performance by improving object detection. We present two novel and complementing models here: Section 3.2 describes first model that is suitable for modeling single object motion, and real-time applications [6]. Section 3.3 describes second model that is useful for learning relationship between concurrently moving object pairs in the scene. The former one is suitable for real-time applications, while the latter is capable of detecting more complex activities. Both of these approaches produce encouraging results on the published data set.

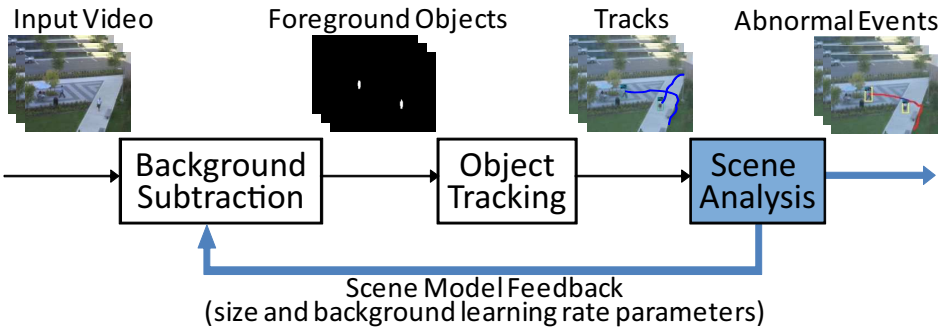


Figure 3.1: Proposed scene analysis approach detects abnormal events and provides scene model feedback. Traditional object detection is improved by using the pixel-level parameter feedback.

3.2 Modeling Single Object Activities

3.2.1 Learning the Scene Model

In this section, we present the details of the structure and learning of the proposed scene model. The visual tracking information serves as the input for our framework. We have used the object detection and tracking system presented in [55]. For a given surveillance video, the tracker produces a set of m tracks $\{T_1, \ldots, T_i, \ldots, T_m\}$, where every track is a set of observations of the same object. For instance, any ith track is a set of n observations $T_i = \{O_1, \ldots, O_j, \ldots, O_n\}$, where $O_j = (t, x, y, w, h)$ contains the time stamp t of observation, location (x, y), width w, and height h of the object. We also use the size (w, h) feature, as it provides useful information for finding anomalous behavior and improving object detection. For instance, this model assists in detecting a pedestrian on the road or a bicyclist on the sidewalk, even when the motion is not very discriminative. Using the set of observations, we want to generate a set of transition vectors that will be used to train the statistical model and provide the details about the motion and size of the objects.

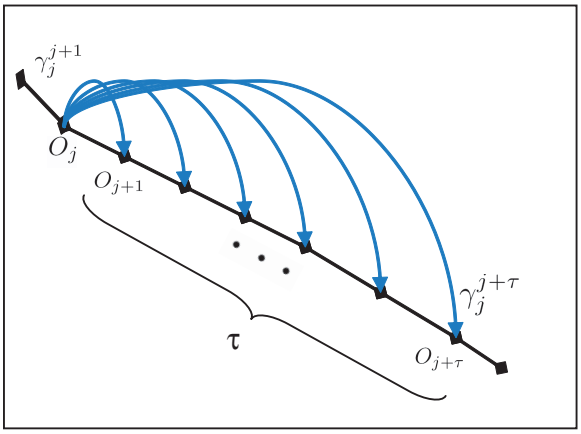


Figure 3.2: A set of observations with transition (blue) vectors connecting them are shown on a synthetic track. O_j and O_k represent two observations of the same object along the track. γ_j^k is the transition vector between O_j and O_k.

For every observation, we compute a set of transition vectors that capture the transition from the given observation to future observations along the same track. Relative velocity is computed for the next observation, as well as a set of subsequent observations. In order to keep the problem computationally tractable, we limit the computation to a temporal window with τ observations. Figure 3.2 shows a synthetic track with marked observations and transition vectors from a particular observation O_j. This provides a means to detect abnormal tracks through the *global* analysis. In many cases mere use of *local* analysis would not be sufficient. One such synthetic example is illustrated in Figure 3.4.

For any observation O_j, relative velocity is computed against all $\{O_{j+1}, \ldots, O_{j+\tau}\}$ to generate a set of transition vectors $\{\gamma_j^{j+1}, \ldots, \gamma_j^{j+\tau}\}$, where transition vector $\gamma_j^{j+\tau} = (x_{j+\tau}, y_{j+\tau}, \tau, w_j, h_j)$. The destination location $(x_{j+\tau}, y_{j+\tau})$ is obtained from the observation vector $O_{j+\tau}$, the duration between the two observations O_j and $O_{j+\tau}$ is τ. (w_j, h_j) represents detected size of the object in

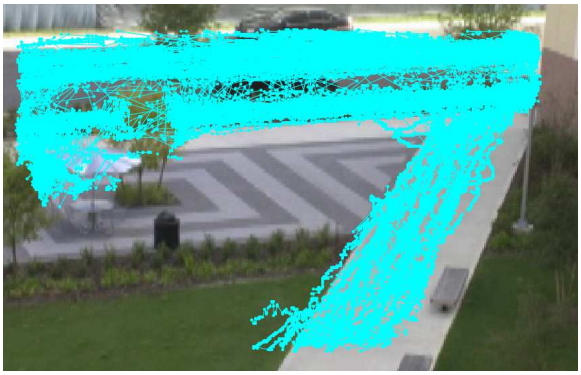


Figure 3.3: A subset of tracks used in the training of the scene model. Multiple transition vectors from each observation contribute towards learning the pdf at that location.

source observation O_j. τ is the length of the temporal window along the track; in the experiments we have used $\tau = 20$.

We model the motion patterns in the scene using the motion and size features, as described above. We use a 5-dimensional random variable Γ_l for every pixel location l, where $\gamma = (x', y', \delta t, w_l, h_l)$ represents one particular outcome of Γ_l. Every transition vector generated from the observations presents a five dimensional random variable. The probability density function (pdf) over this feature space is modeled as a multivariate Gaussian Mixture Model (GMM). This pdf is created for every pixel location in the scene and it models the probability of that location being the source of a transition. The pdf estimated at every location captures the probability of observing an object of a given size which is moving to a specific location in a given duration. The pdf at an intersection of multiple paths can capture the possible transitions in different directions, speeds and sizes of objects.

Learning of the model is performed after a sufficient amount of tracking data has been accumulated. The appropriate duration depends on the amount of traffic in the scene and the required accuracy of the model. For any given location l in the scene, all the observations of the tracks through that location contribute to the pdf at that location. The pdf for the random variable Γ_l is created by utilizing the training instances γ's with l being the source location. The training method described below is repeated for all pixel locations.

A multivariate GMM is used to model the pdf of the random variable Γ_l. The probability of an observation γ belonging to the GMM is given by

$$P(\Gamma_l = \gamma|\theta_l) = \sum_{i=1}^{n} \alpha_l^i p(\gamma|\theta_l^i), \tag{3.1}$$

where n is the number of components detected in the mixture, θ_l^i is the set of parameters defining the ith component with weight α_l^i, and $\theta_l \equiv \{\theta_l^1, \ldots, \theta_l^n, \alpha_l^1, \ldots, \alpha_l^n\}$ defines the complete set of parameters required to specify the mixture model. Each component is modeled as a Gaussian distribution of the form

$$p(\gamma|\theta_l^i) = \frac{1}{(2\pi)^{d/2}|\Sigma_l^i|^{1/2}} e^{-1/2(\gamma-\mu_l^i)^T \Sigma_l^{i\ -1}(\gamma-\mu_l^i)}, \tag{3.2}$$

where d is the dimensionality of the model and $\theta_l^i = \{\mu_l^i, \Sigma_l^i\}$ are the parameters of the model.

The computation of the GMM parameters is performed through an improved Expectation Maximization (EM) based algorithm, which was proposed by Figueiredo and Jain [39]. This particular approach provides a solutions to three major limitations of the basic EM algorithm. First, the number of components does not have to be fixed. This algorithm estimates the number of components by removing the components that are not supported by the data. Second, this approach does

not require careful initialization and starts with a large number of components which are spread throughout the data. Third, this algorithm also avoids convergence towards a singular estimate near the boundary of the parameter space. The details of the algorithm are available in [39], but important points are included here for the sake of completion. The E-step is given by

$$\omega_l^i = \frac{\alpha_l^i(t)p(\gamma|\theta_l^i(t))}{\sum_{j=1}^k \alpha_l^j(t)p(\gamma|\theta_l^j(t))}, \tag{3.3}$$

where ω_l^i captures the conditional expectation of the missing data. $\alpha_l^i(t)$ and $\theta_l^i(t)$ are the parameter values at the iteration t of the EM algorithm. The M-step is given by

$$\hat{\alpha}_l^i(t+1) = \frac{max\{0, (\sum_{m=1}^S \omega_l^i(m)) - \frac{d}{2}\}}{\sum_{j=1}^k max\{0, (\sum_{m=1}^S \omega_l^i(m)) - \frac{d}{2}\}}, \tag{3.4}$$

for $i = 1, \ldots, n$,

$$\hat{\theta}_l^i(t+1) = \arg\max_{\theta_l^i} Q(\theta_l, \hat{\theta}_l(t)), \tag{3.5}$$

for $m : \hat{\alpha}_l^i(t+1) > 0$,

where d is the dimensionality of each mixture component, S is the number of training samples γ used in E-step, and the Q-function estimates the log-likelihood given the current model estimate.

After learning of the complete scene has been performed, the GMM parameters for every pixel location are stored as the scene model. For a given observation, if we only update the pdf of the pixel at the centroid of the bounding box, then the created models could be spatially sparse. To achieve better spatial smoothing of the motion models in the neighboring pixels, we update all the pixels in the bounding box. Note that unlike most of the previous approaches, learning of the proposed scene model does not rely on merging track to estimate the main paths in the

scene. This reduces possible sources of error due to incorrect path estimation or ambiguity of track membership between two or more paths. Another strength of the proposed structure of the scene model is the ability to perform online learning of motion patterns and adaptation to the changing object behaviors in the scene.

3.2.2 Abnormal Behavior Detection

The training phase generates a scene model $\mathbf{\Theta}$ using the observed motion patterns. This model is a set of GMM parameters $\mathbf{\Theta} = \{\theta_l\}$, where l is the location of all the pixels with sufficient training observations. We use this scene model to detect abnormal motion patterns which conflict with the trends observed in the training data. We propose an online approach for detecting anomalies in the latest observation O_t from the test track T. This observation is analyzed as soon as it becomes available after a set of previous observations in the track $T = \{O_1, \ldots, O_{t-1}, O_t\}$. For the task of anomaly detection, *local* and *global* analysis of these observations is performed. In *local* analysis, we conduct the comparison of the current observation O_t with the previous observation O_{t-1} only (first order). This captures many typical anomalies based on instantaneous velocity and size of the detected objects but, it has a limited capability for detecting more complicated anomalies. The *global* analysis, however captures more *complicated* cases by analyzing the current observation O_t with respect to a series of previous τ observations $T' = \{O_{t-\tau}, \ldots, O_{t-1}\}$ (higher order). The transition between any source observation $O_{t-i} \in T'$ and the current observation O_t is defined by the transition vector $\gamma_{t-i}^{t} = (x_t, y_t, i, w_{t-i}, h_{t-i})$, which contains contains destination location, transition time, and the object size at the source location. The pdf $P(\Gamma_{l(t-i)})$ of transition vectors at

the source location $l(t-i)$ from O_{t-i} is used to determine how normal the current transition γ_{t-i}^{t} is. A very low probability value from $P(\Gamma_{l(t-i)} = \gamma_{t-i}^{t})$ is interpreted as representative of an atypical transition. Our goal is to determine if the current observation O_t is abnormal or not by analyzing the trail of observations in the track. Therefore, we use the minimum transition probability

$$\beta_t = \min_i \{P(\Gamma_{l(t-i)} = \gamma_{t-i}^{t})\}, \tag{3.6}$$

for $i = 1, \ldots, \tau$ and the observation O_t is declared abnormal if following condition is true

$$\beta_t < \lambda, \tag{3.7}$$

where threshold λ is applied to the least probable transition. This provides a means of detecting atypical transitions that originated from any one of these higher order transitions. Hence, both local and global anomalies can be detected through this framework. Our approach performs online analysis of the motion patterns to detect anomalies as soon as they occur.

We use this framework to detect various types of anomalous behaviors. Figure 3.5 presents various types of detected anomalies in a real video. These include pedestrians on the road and grass, skateboarder and bicyclist on the sidewalk, pedestrians sitting down, etc. In addition, we can also catch anomalies like violations of one-way traffic, which is important on the road and in some airport hallways. Figure 3.4 presents a synthetic scene to illustrate the case of global anomalies. Randomly generated tracks (Figure 3.4(a)) were used for training completely follow one of the four paths. Our goal is to detect the tracks whose behavior is normal locally but not globally. This is important, for instance at the airport where pedestrians from one path are not allowed to switch to another intersecting path. Another example could be of cars that are not

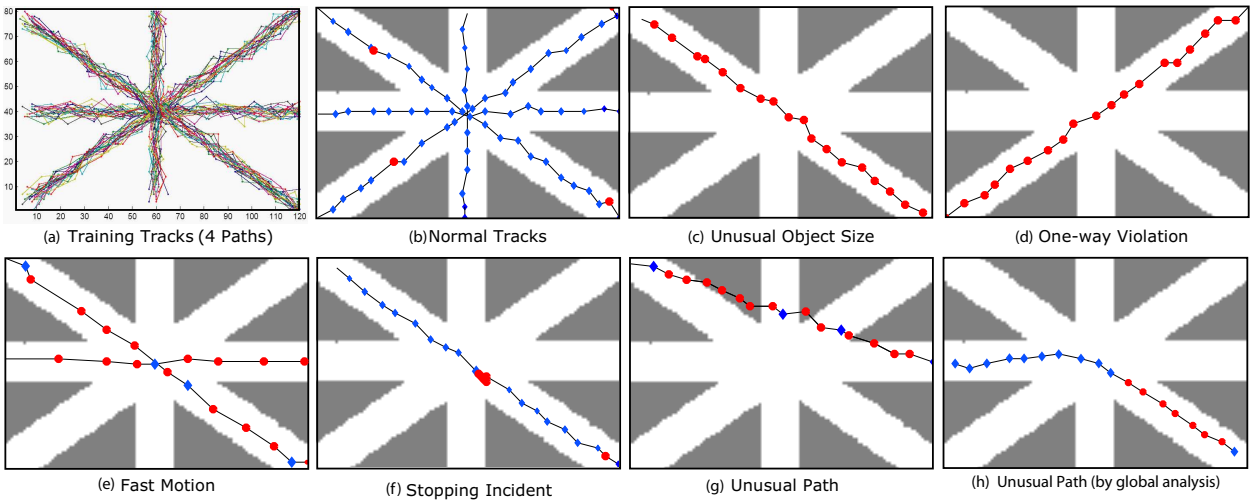


Figure 3.4: Global anomaly: when the tracks are not allowed to change paths, global analysis detects the violations. Every observation is labelled either normal (blue diamond) or abnormal (red circle). Gray background is the region without motion model. (a) Training set of random unidirectional tracks (along four paths). (b) Local analysis fails to identify anomaly, while (c) global analysis highlights the observation that take an unusual path.

allowed to turn on an intersection. Figure 3.4(b) and (c) show the outcome of the local and the global analysis respectively. Local analysis the first order transition between observations is not sufficient to detect such anomalies. Instead we use higher order transitions to capture the global structure of the track. This type of analysis can also be useful for detecting cyclic motion or repeated U-turns which can be abnormal.

3.2.3 Improving Object Detection

An important application of the proposed scene modeling approach is to improve object detection utilizing the patterns in the observed tracks. The knowledge of object parameters (size and

speed) at every pixel location is used for this purpose. There are certain components in traditional background subtraction algorithms [99, 38] that could benefit from this scene knowledge. These parameters are traditionally considered consistent throughout the scene, but this limits the performance of object detection. The scene model provides the feedback information (see Figure 3.1) for every pixel to update the parameter values according to the scene information. The use of the proposed scene model is presented in the following for two parameters, minimum object size and background learning rate.

3.2.3.1 Minimum Object Size

The minimum size (s) of the detected objects is the first parameter which benefits from our scene model. Size s is defined as the area of the blob detected after background subtraction. If this value is set too high, then detection of valid small objects in the far view camera fails. On the other hand, if this value is too low, then some noisy segments and broken parts of larger object blobs are reported as separate objects. Instead of a fixed global value for the parameter s, we present a method for automatically obtaining the appropriate value of the s parameter at different pixels.

In order to improve the accuracy of object detection, we use the proposed scene model to estimate the probability of observing an object of a given size at the current location. In the learnt scene model, the pdf at every pixel location captures the joint probability of motion and size. For size-based analysis, we extract the marginal pdf for the size parameters

$$P(w,h) = \sum_{x=1}^{m}\sum_{y=1}^{n}\sum_{t=1}^{\tau} P(x,y,t,w,h), \tag{3.8}$$

where n rows & m columns is the size of the image and the maximum transition duration modeled in the pdf is τ. As mentioned in [63], this marginal pdf for $\mathrm{x_{wh}} = (w, h)$ can be represented as

$$P(\mathrm{x_{wh}}) = \sum_{i=1}^{n} \alpha_i p(\mathrm{x_{wh}}|\theta_i^{\mathrm{wh}}), \tag{3.9}$$

where θ_i^{wh} represents the parameters for ith bivariate Gaussian with mean μ_i^{wh} and covariance Σ_i^{wh}

$$p(\mathrm{x_{wh}}|\theta_i^{\mathrm{wh}}) = C \exp\{-\frac{1}{2}(\mathrm{x_{wh}} - \mu_i^{\mathrm{wh}})^T \Sigma_i^{\mathrm{wh}} (\mathrm{x_{wh}} - \mu_i^{\mathrm{wh}})\}, \tag{3.10}$$

where

$$C = \frac{1}{2\pi |\Sigma_i^{|\Sigma_i^{\mathrm{wh}}}|^{1/2}},$$

$\Sigma_i^{|\Sigma_i^{\mathrm{wh}}}$ is Schur's decomposition of Σ_i with respect to Σ_i^{wh}, and Σ_i is 5×5 covariance matrix from original joint pdf.

The marginal pdf is created at every pixel location and it captures the density of observed object sizes at that location. For illustration purpose, we use this pdf to generate the size map shown in Figure 3.8. The mean value of width and height from the Gaussian component with highest weight is used in the computation of the most probable size at a given pixel location. This value of size is used as the intensity of the corresponding pixel location in the size map. Note that the size values on the road region are much higher than those on the sidewalks. The size values can be observed to be gradually reducing as the objects move away from the camera.

The parameters of the marginal pdf at every pixel are passed to the object detection module as feedback. Figure 3.1 shows the feedback flow of the pixel level parameters representing the size pdf at each pixel. The background subtraction algorithm generates a set of foreground blobs of

different sizes. For each of the foreground blob at location (i, j) with size (w, h), we compute the probability $P(w, h)$ using the marginal at (i, j). A very low value means that the current blob is most likely a false observation. Suppressing valid objects at unexpected locations can be avoided by defining the s parameter at the current location as

$$s = s_{min}P(w, h) + s_{max}(1 - P(w, h)), \tag{3.11}$$

where $[s_{min}, s_{max}]$ specify the range for s value. This range does not greatly affect the sensitivity of the detection module. In our experiments we used $[50, 150]$ range for two different scenes. Pixels locations with missing models or unexpected object size produce low probability values, which generate a high s value for that pixel. This approach assures that very small noisy observations are not approved as valid objects. High probability values result in small s value which assures that even small sized valid objects are not missed. This provides a means for the object detection module to have different s values for different pixels based on the learnt scene model.

3.2.3.2 Background Learning Rate

The background learning rate (ρ) is used to update the learnt background model in order to adapt to slow changes in the scene [99]. For instance, if a table is moved in the room, the new setting is learnt as a part of the background. However this feature can cause a problem when the goal is to consistently track an object that briefly becomes stationary. For instance, if a car stops briefly on a traffic light, it can be quickly learnt as a part of the background if ρ is too large. On the other hand if ρ is too small then the valid changes in the scene would not be incorporated in a suitable time.

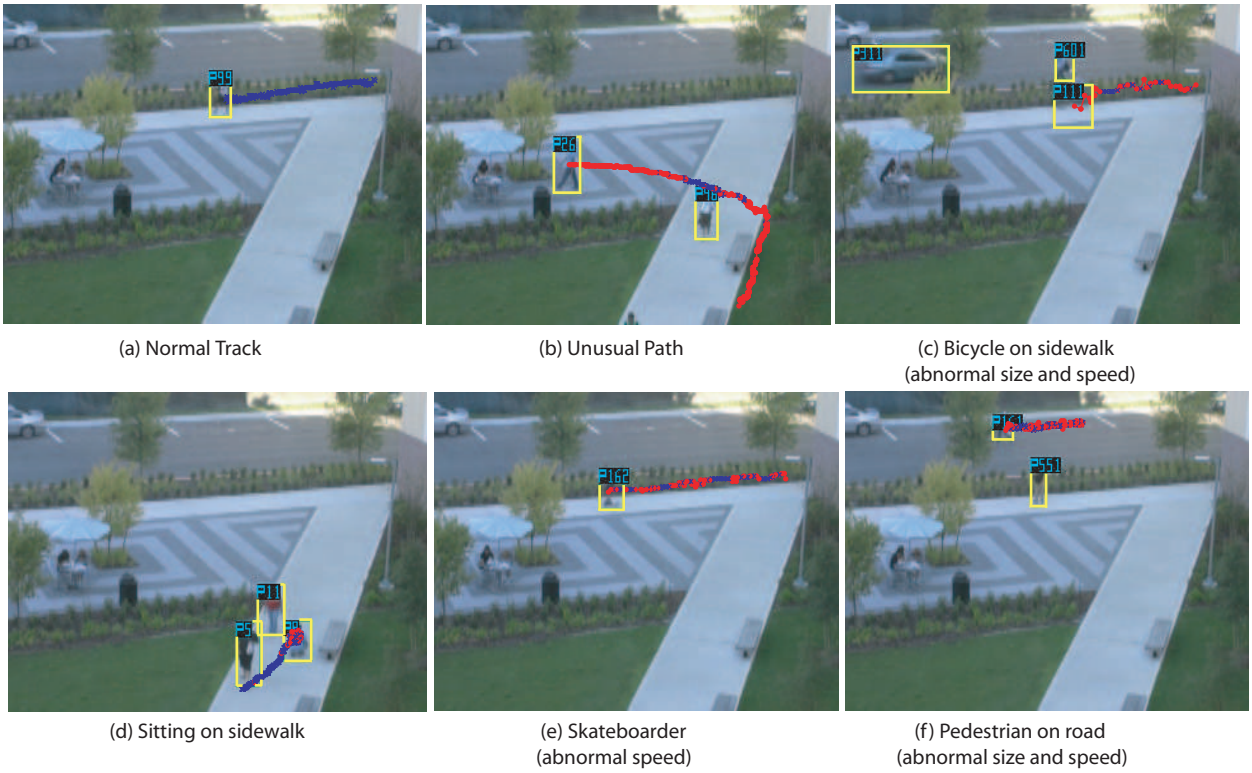


Figure 3.5: Scene 1. Detected abnormal observations are labelled red and normal observations are blue. (a) All normal observations of a typical pedestrian (b) The pedestrian follows an unusual path. (c) The observations of a bicyclist are also classified as abnormal, because of the abnormal speed and size of the object. (d) A person stops in the middle of the sidewalk and sits down. Note that the observations were correctly labelled normal before the person sat down. (e) A skateboarder, whose observed size is the same as that of the pedestrian but the speed helps in distinguishing them. Some of the observations are detected normal because of only a slight difference in speed. (f) Unusual size and speed prove to be useful in case of a pedestrian walking on the road. All of the above mentioned tracks are part of the testing video, which is different from the training video.

This dilemma suggests that we locally tweak the value of ρ depending on the behavior of objects in the scene.

The proposed scene model captures different speeds at a particular location. We identify the regions in the scene where objects become stationary, including the exit zones. The learning rate is lowered only for the pixels belonging to these regions. Similar to the approach for the minimum object size, we extract the marginal pdf that captures the motion information. The marginal pdf

$$P(x, y, t) = \sum_{w} \sum_{h} P(x, y, t, w, h), \tag{3.12}$$

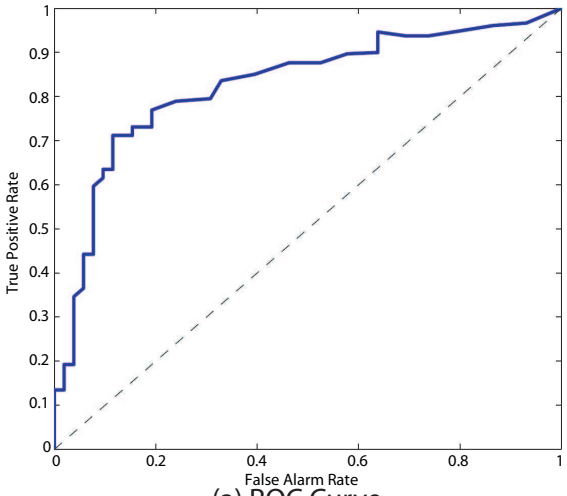
is extracted at every pixel. The GMM component parameters are updated in a manner similar to the minimum size. The object detection could fail because of the high ρ value, therefore we identify the regions where objects stop and reduce ρ. This is done by analyzing the smallest object speed ($\hat{v}$) captured at every pixel. The difference between pixel location and the GMM component mean is used to compute this speed. The interpolated value of ρ can be computed using following expression

$$\rho = \rho_{min} P_v(\hat{v}) + \rho_{max}(1 - P_v(\hat{v})), \tag{3.13}$$

where P_v is a zero mean normal distribution used to signify reducing speed, and $[\rho_{min}, \rho_{max}]$ are the two extreme values of the learning rates to be used. The aim for this formulation is to automatically choose a value of ρ for every pixel depending on the type of object behavior observed during the training phase.

3.2.4 Experimental Results

The performance of the proposed framework was tested on real sequences captured from three different surveillance cameras. A typical scene observed from the first camera is shown in Figure 3.5.



(a) ROC Curve

	Training	Testing
Normal Tracks	1342	217
Abnormal Tracks	0	31

(b) Ground truth

Figure 3.6: Anomaly detection performance on the scene shown in Figure 3.5. (a) ROC curve for the 30 mins test video. (b) Table with ground truth number of tracks used in training and testing.

Realtime object detection and tracking was performed using the UCF KNIGHT system [55]. Initial training is performed off-line and testing for anomalous behavior detection was performed using the tracking results from a 30 minute test video. Figure 3.6(b) shows the details of the training and testing sets used for this experiment. Matlab implementation runs at approximately 26 fps for this module on a 3GHz Pentium D PC machine. Figure 3.5 presents the output of abnormal behavior detection in the test sequence.

The proposed approach declares an observation abnormal as soon as it is received from the tracker. Figure 3.5 shows a set of detected abnormal behaviors in addition to a normal track. The first one is an unusual path, where a pedestrian is tracked through a region where not enough training tracks were observed. Next, a bicycle is on the sidewalk, which was not present in the training video. The unusual speed and size of the bounding box provides evidence of such anoma-

lies. Another similar anomaly (e) shows a skateboarder going faster than pedestrians. Most of the observations are labelled as abnormal even when the observed size is very similar to that of a pedestrian. (d) shows a case where a pedestrian sits down on the sidewalk and (f) shows a case where a pedestrian is detected on the road. This particular anomaly is captured by difference in speed and size of the observed object and the scene model. The results show only a small number of observations are misclassified. The majority decision for the complete track keeps the results accurate. Figure 3.6(a) presents the ROC curve depicting the accuracy of anomaly detection.

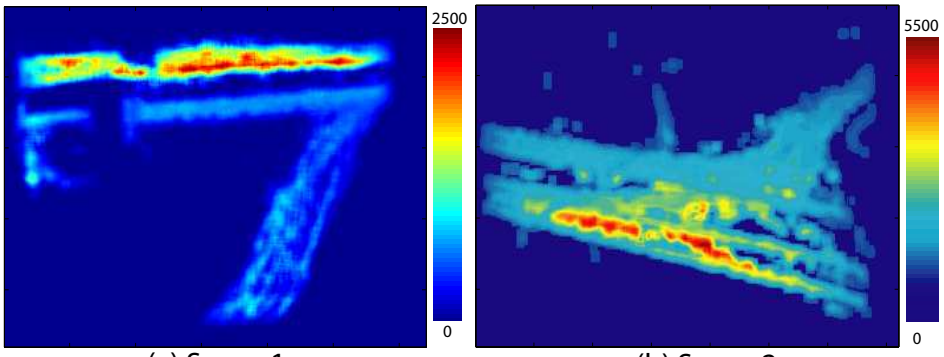
Figure 3.8(a) presents the object size map extracted from the learnt scene model scene 1 shown in Figure 3.5. The high intensity values along the road are generated by the vehicles. As the objects move away from the camera the observed sizes reduce, which reflects here as reducing intensities along the sidewalk. Similarly, Figure 3.8(b) shows the size map for scene 2 shown in Figure 3.7.

The experiments of improving object detection are performed on video from two other surveillance cameras. Results of the improvement in the object detection using the size parameter feedback are presented in Figure 3.7. Two real scenarios are shown here that support the claim that the proposed size map outperforms the case with fixed s value. In the case of (b), the lowest value of $s = 50$ is chosen and in both scenarios, false positive objects are detected. In the first scene, a small broken part of the pedestrian's shadow is detected as a valid object and in the second case, a noisy observation on the lamp post is declared as a valid object. In the case of (c), a comparatively higher value of $s = 150$ is chosen and it clearly misses the pedestrians that are farther away from the camera. Finally, (d) presents the improved object detection using the proposed size map which provides a different s value at each pixel location. All the actual objects are detected without any

Figure 3.7: Scene 2. Improvement in object detection by the proposed size model. Each row presents an instance in the same video. Column (a) shows the manually extracted patches of the objects currently present in the scene. Column (b) is the output when a uniform global value of $s = 50$ is used. Noisy foreground blobs are also detected as valid objects (red ellipses). (c) presents output when $s = 150$ is used throughout the scene. Individuals are not detected (red ellipses) when the object size is small. (d) presents results of the proposed size model. In both scenarios the valid objects are detected and the noisy observations are avoided.

noisy detections. The automatically learnt size map proves to be very useful in accurately capturing the perspective distortions in the scene.

Figure 3.9 presents results of automatic feedback for pixel-wise update of the background learning rate. This camera covers an intersection with traffic lights where cars may stop up to approximately 40 seconds. The scenario shown in this figure contains a black car arriving, stopping for a red light, and then driving away. Figure 3.9(a) shows the output using a typical value of



(a) Scene 1 (b) Scene 2

Figure 3.8: The object size maps are computed for scene 1 (Figure 3.5) and scene 2 (Figure 3.7). Intensity at every pixel location is the most probable size of the object observed at that location. The highest intensity is observed for the vehicles along the road. Note the gradually reducing sizes due to perspective effect.

learning rate (ρ = 0.01). The target of continuously tracking the stationary car could be achieved by increasing ρ, but this can induce spurious detections where the background changes rather quickly. Using the proposed parameter feedback approach, we can isolate this increase of ρ to only the regions where it is required (i.e. where traffic stops). In the experiments, we have used $[\rho_{min}, \rho_{max}] = [0.005, 0.1]$ as the extreme values of the learning rate. Figure 3.9(b) shows the detection output by using the proposed feedback approach for learning rate. The new detection through this approach have been highlighted.

3.3 Modeling Object Pair Activities

The approach presented above models the motion patterns of each object independently. The scene model accumulates observations from multiple tracks but each sample in the pdf represents

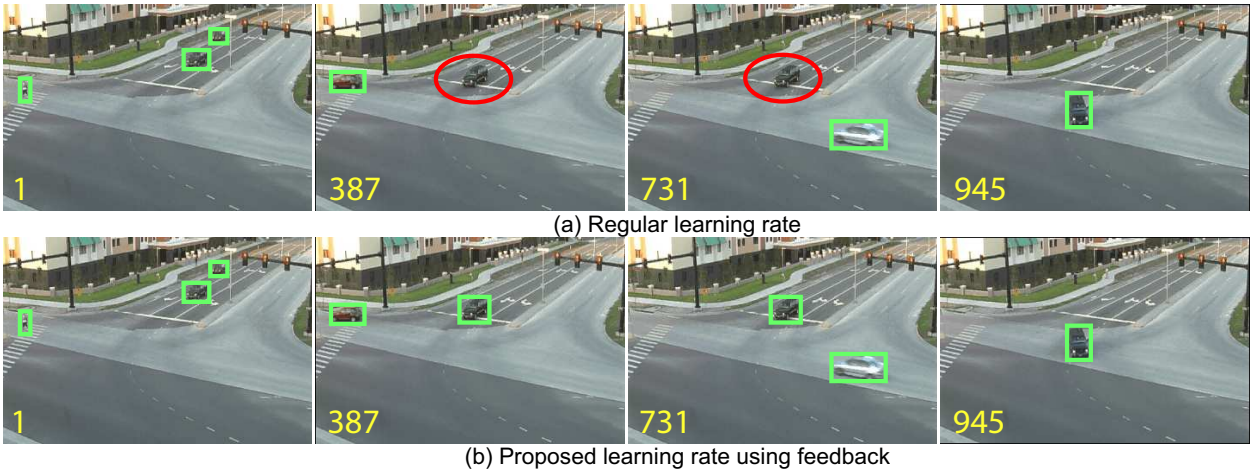


Figure 3.9: Scene 3. Improvement in object detection using the proposed feedback approach for updating learning rate. Video sequence progresses from left to right. (a) Using the uniform background learning rate (ρ = 0.01) for the whole scene. (b) Detection results using the proposed approach for updating background learning rate. Red ellipses highlight the car that was not detected by the regular approach but was later detected by our approach.

motion of a single object independent of others. This type of model lacks the ability to capture interactions between two or more objects. For instance, when a car drops off a person, there is useful information in the two tracks about the mutual interaction of these objects. We are interested in exploring the possibility of creating a statistical model of pairs of objects that are concurrently observed in the scene. This would complement the approach proposed above by adding the ability to model object interactions. Such a model will be able to capture the functionality of the current single object model, as well as the new functionality of modeling object pairs. This could prove to be useful in detecting more complex abnormal behaviors, such as illegal drop-off/pickup, traffic light violations, etc. In this section, we present a new composite model that captures the interaction

of object pairs in the scene to detect such behaviors. We show that the novel model presented here is useful for identifying abnormal object interactions, as well as single object anomalies.

3.3.1 Learning the Scene Model

The observation vector (x, y, w, h, τ) of each object consists of bounding box centroid (x, y), width (w), and height (h) along with the time of observation (τ). For a pair of objects a and b tracked concurrently in the scene, we can build a composite transition vector

$$\gamma_{a,b} = (\gamma_a, \gamma_b), \tag{3.14}$$

$$\gamma_a = (x_a, y_a, x'_a, y'_a, w_a, h_a, \tau_a), \tag{3.15}$$

$$\gamma_b = (x_b, y_b, x'_b, y'_b, w_b, h_b, \tau_b), \tag{3.16}$$

where γ_a represents the transition of an object a from a source location (x_a, y_a) to a destination location (x'_a, y'_a) in time τ_a. Similarly, γ_b represents the transition of object b. Note that τ_a and τ_b could be different if considering two transitions of different degrees. If one of the objects is occluded for a few frames, two different transition times can be used. This scenario is handled seamlessly by the model. Figure 3.10 illustrates several transition between three objects, a, b and c, concurrently present in the scene.

The composite transition vector holds the semantically holds the commutative property, (i.e. $\gamma_{a,b} = \gamma_{b,a}$). In order to reduce the complexity of the KDE, we only use one of the two possibilities ($\gamma_{a,b}$ or $\gamma_{b,a}$). Let Γ be a 14-dimensional random variable whose observations are $\gamma_{a,b}$. We use KDE to learn the probability density of this random variable. A multivariate distribution $p(\Gamma)$ is

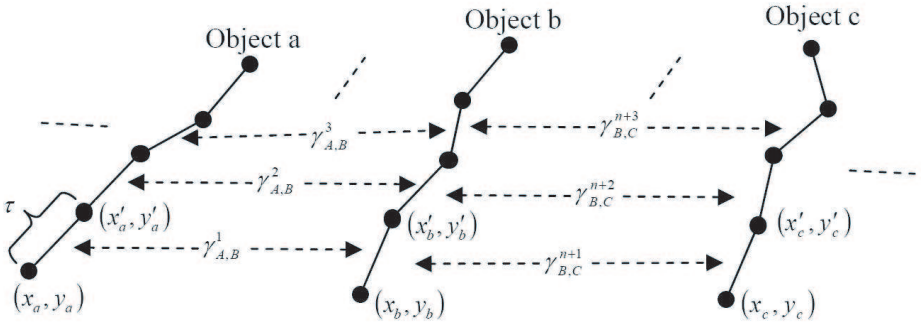


Figure 3.10: Modeling track interaction between the objects tracked concurrently.

formed through the set of joint object transitions $\gamma^1_{a,b}, \gamma^2_{a,b}, \gamma^3_{a,b}, \cdots$ observed during the training period. The estimated probability density is computed as

$$\hat{p}(\Gamma = \gamma) = \frac{1}{n\sqrt{H}} \sum_{i=1}^{N} K(\frac{\gamma - \gamma^i}{\sqrt{H}}), \tag{3.17}$$

where H is a symmetric positive definite dxd bandwidth matrix, and K is a 14-dimensional kernel function. In our experiments we use a Gaussian kernel. The bandwidth of the kernel is one of the parameters that can affect the accuracy of the model. It is estimated through the minimization of the mean-squared error between the estimated and the real density $[\hat{p}_H(\gamma) - p_H(\gamma)]^2$. We use a likelihood-based search for bandwidth selection, see [105] for more details.

The proposed model represents the joint transition of an object-pair concurrently observed in the scene. We can also obtain a derived model to test a single object transition (say γ) in the scene. This is done through obtaining marginal distributions of the two parts of the learnt pdf, given by

$$\hat{p}(\Gamma_a) = \int_b \hat{p}(\Gamma_{a,b}) db, \tag{3.18}$$

$$\hat{p}(\Gamma_b) = \int_a \hat{p}(\Gamma_{a,b}) da, \tag{3.19}$$

Table 3.1: Algorithm of mean shift based local refinement to estimate best joint transition vector.

Objective: Given the initial joint transition vector $\gamma_{a,b}$, use mean shift to compute a refined transition vector $\hat{\gamma}_{a,b}$. **Algorithm:**

1. Let $\gamma^t_{a,b} = \gamma_{a,b}$.
   for $t = 1$ to T do
   (a) Generate a set of N samples by using $\mathcal{N}(\gamma^t_{a,b}, \Sigma^t_{a,b})$ and compute mean state γ_m by Equations 3.22 and 3.23.
   (b) If $\|\gamma^t_{a,b} - \gamma_m\| \leq$ threshold
   then break for
   otherwise let $\gamma^t_{a,b} = \gamma_m$.
   end for
2. Refined estimate of the transition vector $\hat{\gamma}_{a,b} = \gamma^t_{a,b}$.

where $\Gamma_{a,b}$ is the full 14-dimensional random variable with the transition vectors of both objects a and b, Γ_a is the random variable with first 7-dimensions of the original composite transition vector $\Gamma_{a,b}$ and Γ_b is the random variable with last 7-dimensions of the original composite transition vector $\Gamma_{a,b}$. The representative model for the single object transition γ is then selected by choosing the best candidate: $max(\hat{p}(\Gamma_a = \gamma), \hat{p}(\Gamma_b = \gamma))$. Such a representation for the single object transitions holds when the goal is to identify the outliers from the learnt model.

3.3.2 Abnormal Behavior Detection

We define the behavior of an object as a set of transition vectors generated from the track. Each of the transition vectors can be tested against the learnt pdf individually, as well as a part of the joint transition vector with other objects being observed. For a comprehensive test of a given object, we perform both individual and joint test for a transition using the learnt pdf. In principal, we could identify the outlier by applying a threshold on the computed probability density against the object transition under consideration. However, in practice, the higher dimensionality of the pdf makes it sensitive to noise and sparsity. To address this problem, we propose a local sample refinement step based on the principal of mean shift [31]. For a given joint transition vector $\gamma_{a,b}$ between objects a and b, a refined transition vector $\hat{\gamma}_{a,b}$ is estimated through an iterative approach summarized in Table 3.1. We start with generating a set of N samples from a normal distribution $\mathcal{N}(\gamma^t_{a,b}, \Sigma^t_{a,b})$ around $\gamma^t_{a,b}$ with covariance $\Sigma^t_{a,b} = diag(\varepsilon^t_{a,b})$, where $\varepsilon^t_{a,b}$ is the joint transition error. This error can be computed through the mean of the absolute error $\Delta\gamma^t_{a,b}$ at every iteration t of the refinement algorithm as follows:

$$\Delta\gamma^t_{a,b} = |\gamma^t_{a,b} - \hat{\gamma}^t_{a,b}|, \tag{3.20}$$

$$\varepsilon^t_{a,b} = \frac{\Delta\gamma^t_{a,b} + \Delta\gamma^{t-1}_{a,b}}{2}. \tag{3.21}$$

Similar to the mean shift algorithm [31], we use N weighted samples γ_i in the neighborhood of the original sample γ. The refined mean at every iteration is computed as

$$\gamma_m = \frac{\sum_{i=1}^{N} G(\gamma - \gamma_i) w(\gamma_i) \gamma_i}{\sum_{i=1}^{N} G(\gamma - \gamma_i) w(\gamma_i)}, \tag{3.22}$$

Table 3.2: Algorithm for abnormal behavior detection in object pairs.

Objective: Given N objects in the scene, identify set abnormal object behaviors (set A).

Algorithm:

1. Initialize $V_{N\times N}$ voting matrix to 0.
2. Populate voting matrix for all combinations of objects

   for $i = 1$ to N do

   for $j = i$ to N do

   (a) if $\hat{p}(\gamma_{i,j}) >$ threshold (using Equation 3.25)

   then $V_{i,j} = V_{i,j} + 1$

   otherwise $V_{i,j} = V_{i,j} - 1$

   end for

   end for
3. Identify abnormal behavior

   A = ø

   for $i = 1$ to N do

   (a) $positive\_counts : count(V_{i,N} > 0) + count(V_{N,i} > 0)$

   (b) $negative\_count : count(V_{i,N} < 0) + count(V_{N,i} < 0)$

   (c) if $negative\_count > postive\_count$

   then $A = A \cup i$

where we use normal distribution as the kernel around each sample. In addition, weight w computed from the density is used in order to include the likelihood of each sample γ_i based on the training data, and is defined as

$$w(\gamma) = \hat{p}(\Gamma = \gamma). \tag{3.23}$$

The refined joint transition vector $\hat{\gamma}_{a,b}$ can now be used to detect abnormal behavior of the single object transitions $\hat{\gamma}_a$ and $\hat{\gamma}_b$, by utilizing the marginal densities as explained in the last section. However, in order to effectively utilize learnt scene model for anomaly detection among object pairs, we have to evaluate both density using the combinations of object pairs:

$$\hat{p}(\Gamma = \hat{\gamma}_{a,b}) \equiv max(\hat{p}(\Gamma = \hat{\gamma}_{a,b}), \hat{p}(\Gamma = \hat{\gamma}_{b,a})). \tag{3.24}$$

This may increase the computational complexity of the anomaly detection step. Another option is to build a more complex model by using both forms $\gamma_{a,b}$ and $\gamma_{b,a}$ as a part of the training data. This implies increasing the amount of training data in the KDE model by a factor of two, which would also significantly increase the complexity of density computation as given in Equation 3.17.

In order to decide whether a particular object is presenting a normal or abnormal behavior, we can use the history of the object to handle noise and consolidate the decision over the life of the track. This is done by fusing the computed probability densities through a Markov Chain. In the case of the track pairs, the duration considered would be the duration of frames during which both of the objects are observed. Let $\gamma^f_{a,b}$ be the joint transition vector at frame f

$$\hat{p}(\Gamma = \gamma^{F2}_{a,b}) \equiv \prod_{f=F1}^{F2} \hat{p}(\Gamma = \gamma^f_{a,b}, \tag{3.25}$$

where $[F1, F2]$ is temporal interval where the two tracks co-exist. In the case of evaluating a single track anomaly, this interval can be the full or partial duration of the track.

The final stage involved in deciding the abnormal behavior in the presence of multiple objects is the consolidation of decisions from several object pairs into a final decision. The main purpose is to declare an object presenting abnormal behavior when most of the objects in the scene support

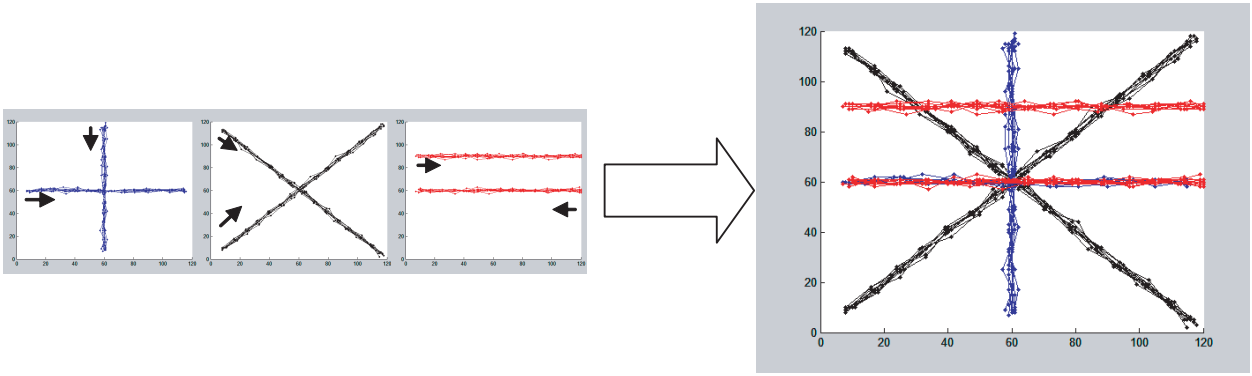


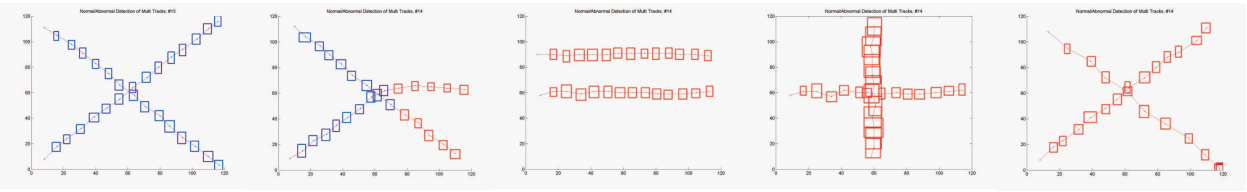
Figure 3.11: Synthetic scene with three pairs of interacting paths (group of tracks) generated to train the scene model. The small arrows show the direction of motion along the path.

the decision. Such an approach is useful in making a robust decision that could otherwise be misleading due to measurement noise or lack of training data. We propose to use a simple majority voting scheme for this purpose, where all the combinations of object pairs are evaluated to make a decision about object behavior.

3.3.3 Experimental Results

In this section we present the results of the experiments performed on the toy example using synthetic data, as well as two real scene with various abnormal behaviors.

The synthetic data used to create a KDE model is shown in Figure 3.11. The tracks generated for this experiment were randomly generated from a normal distribution with pre-specified parameters for the respective path. The KDE model is generated from these training tracks using the approach mentioned in Section 3.3.1. The test data used to demonstrate the results contains one normal event and four abnormal events, which include unusual path, unusual direction of motion,



(a) (b) (c) (d) (e)

Figure 3.12: The sample output of testing pairs of synthetic object tracks is shown here in case of: (a) normal pair of tracks, (b) one track with unusual path, (c) unusual tracks in opposite direction to the training tracks, (d) a track with unusual size, and (e) a track with unusually high speed.

unusual object size, and unusually high speed. Figure 3.12 presents the output of the test phase where the objects pairs are used for testing. The observations of the tracks labeled as normal are shown in blue, while the observations labeled as abnormal are in red. The objective here is to analyze the test tracks in light of the training tracks and identify the parts of the tracks that deviate from the normalcy model. Note that in Figure 3.12(b) both the tracks have been labeled abnormal after one of the object takes the unusual path. This shows that by using the object pairs we can identify the relationship between objects. The independent decision based on single object will be eventually used to identify the only object which is abnormal out of the two.

In case of the real scene, we recorded videos from two different sites. Video from scene 1 is 4 hours and 30 minutes long while that from scene 2 is 2 hours 30 minutes long. Each video is divided into training and testing portions. There were 1616 tracks in scene 1 that were used for training, while 193 tracks were used for testing, with 11.94% testing to training set ratio. Similarly, 925 tracks in scene 2 were used for training, while 77 tracks were used for testing, with 8.32% training

(a) Unusual person pickup event

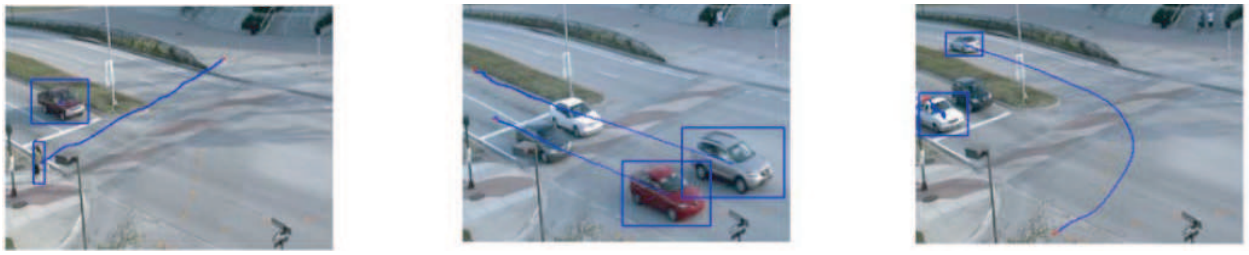
(b) Unusual person drop off

(c) Close proximity of moving person and vehicle

Figure 3.13: Sample results of anomalous behavior detection. Normal and abnormal detections are shown in blue and red, respectively.

to testing set ratio. Training was performed using the approach presented in Section 3.3.1. The KDE model was created using the likelihood-based bandwidth selection approach.

The events used for testing in scene 1 include person drop off by vehicle, person pickup by a vehicle, and close proximity of moving person and vehicle. The results of scene 1 are shown in Figure 3.13. The portions of the tracks highlighted in red are labeled as abnormal, while the portions in blue are labeled as normal. We are detecting the vehicle drop off and pick up events as abnormal because there were not many examples of this in the training data in those regions. The third case of close proximity is particularly interesting because it re-emphsizes the importance of

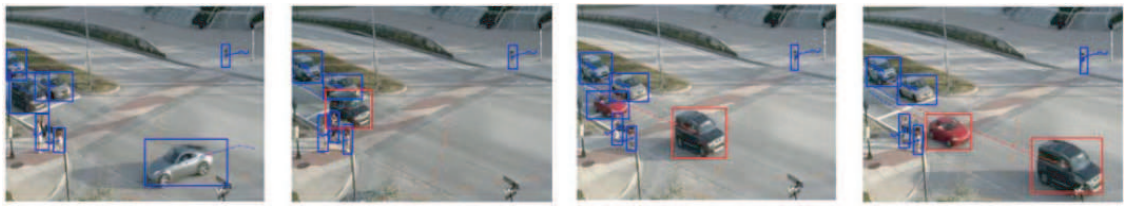


(a) (b) (c)

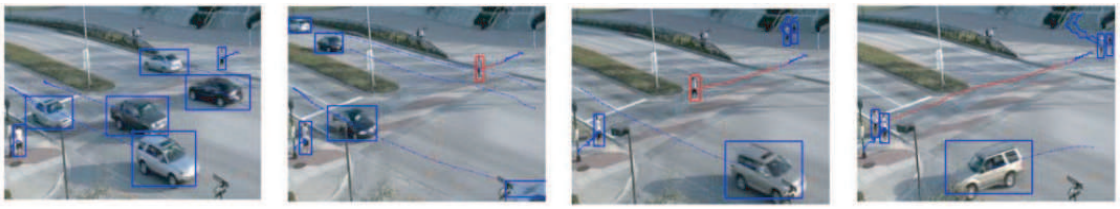
Figure 3.14: Examples of correctly detecting normal events. (a) Person on a crosswalk, (b) vehicles driving straight, and (c) vehicle turning.

using the proposed model to detect dangerous situations on the road. Another interesting aspect of this event is that it was detected as normal when using only the single track model. There were quite a few examples of pedestrians jaywalking in that region and it was learnt as normal behavior. If the vehicle and the person are analyzed in isolation, they are detected as normal, however we are able to identify unusually close proximity by using the object pair model.

Similarly, there are some interesting events in scene 2 that have been identified by the proposed approach as abnormal. We first present some of the normal events in the scene, as shown in Figure 3.14. The person crossing the road on the crosswalk, while the vehicles are stationary, is correctly detected as normal. Similarly, vehicle following the usual traffic pattern learnt in the scene are also labeled as normal. Figure 3.15, however shows examples of the abnormal behavior detected in this scene. The first event is the violation of the red traffic light. One can notice other vehicles still parked while the red and black cars go through the traffic light. It is possible that the stationary cars would have moved late. If that happens while the violating car is still in the field of view, then the decision has a chance to be changed provided there is sufficient time

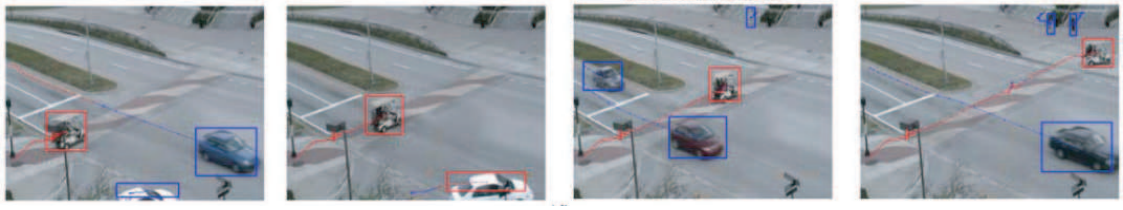


(a) Red light violation by black and red cars.

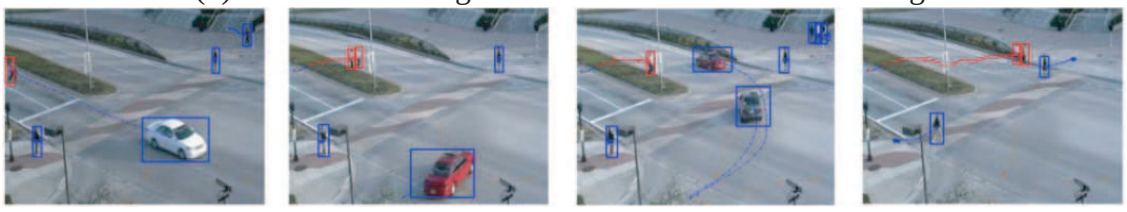


(b) Two individuals jaywalking.

(c) Two individuals jaywalking.



(d) Golf-cart crossing the road while vehicles are moving.



(e) Two individuals jaywalking.

Figure 3.15: Examples of anomalous behavior detection

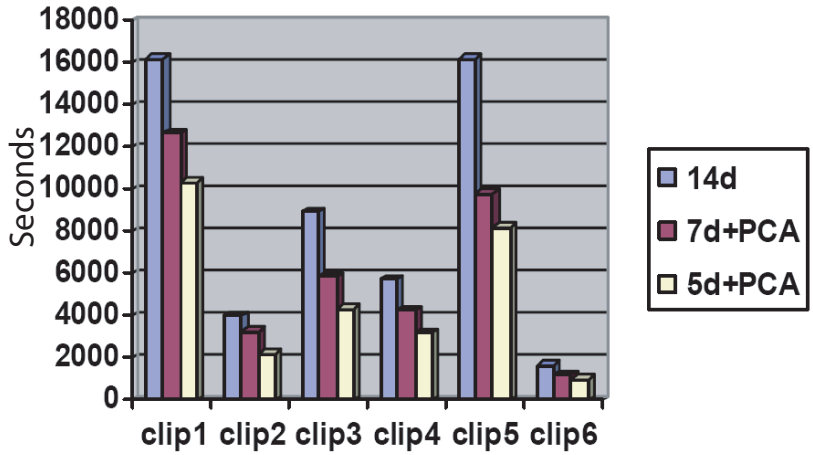


Figure 3.16: Runtime comparison of anomaly detection when using three variants of the KDE model: the original 14-dimensioal KDE, reduced to 7-dimensions after PCA, and reduced to 5-dimensions after PCA.

available. Other than this, there are examples of people jaywalking either on the crosswalk or away from it. The case of walking on the crosswalk when it is not allowed is quite interesting and shows the effectiveness of the proposed object pair model. When only single object model is being used, such subtle anomalies are missed because the relationship between objects' behavior is not captured. Notice that a golf-cart is also detected crossing the road illegally. There are two reasons for this event being detected as abnormal. First, the other vehicles on the road are still moving when the golf-cart crosses the road. Second, the size of the golf-cart is unusual for this part of the scene because we only typically observe humans crossing the road in that region.

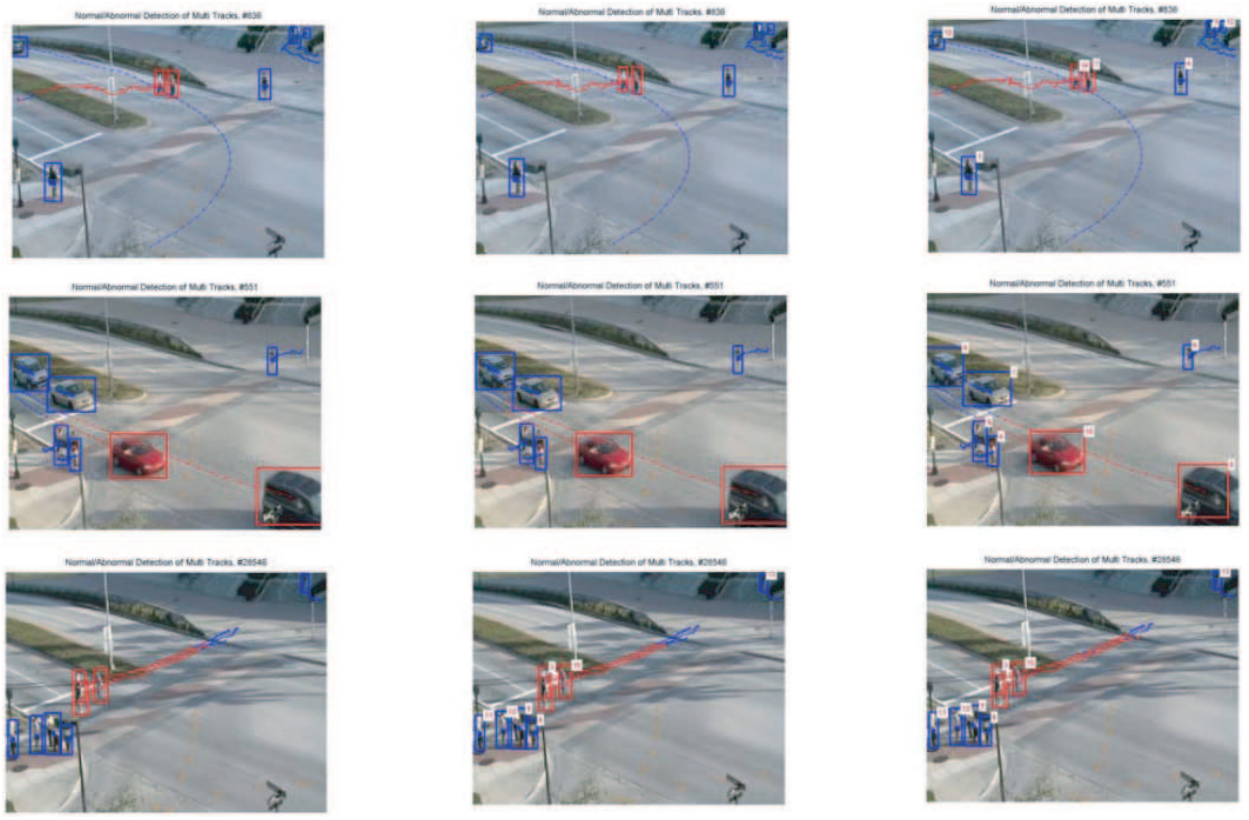
We performed an experiment to study the issue of the high dimensionality of the model. The originally proposed KDE is 14-dimensional and there are hundreds of thousands of samples that are stored in the model. This results in a high computation cost when this model is used for

computing kernel bandwidth, as well as the probability density for a test samples. To address this problem, we present the use of reduced dimensionality through the use of principal component analysis (PCA). This is done at the training stage when the training data is used to evaluate the principal components. We have experimented by reducing the dimensionality to first 7 and then 5 dimensions. The 14-dimensional test samples are then projected into the reduced feature space using the principal components obtained during training. We noticed a significant speedup in the performance of both the bandwidth selection and the testing stage. Figure 3.16 shows the reduction in the runtime when using 5 and 7 dimensions, as compared to the original 14-dimensional feature space. In the 5-dimensional case the speed is almost double. This speedup is achieved without the loss of performance accuracy, as shown in Figure 3.17. The three abnormal events shown earlier are still correctly identified when using the reduced 7 or even 5 dimensions.

3.4 Summary

In this chapter we have presented two novel approaches for coarse level activity modeling in a scene. The first approach models and learns the motion patterns of individual objects in the scene, while the second one also models the interactions between objects pairs. While the first approach is more suitable for lightweight real-time applications, the second one is more powerful for detecting relatively more complicated and useful behavior in a scene.

In the first approach, we adopt an unsupervised learning based approach, which models object motion and size at every pixel location. The proposed framework provides a means of performing higher level analysis to augment the traditional surveillance pipeline. The pdf of motion patterns at



(a) (b) (c)

Figure 3.17: Anomaly detection performance comparison between different model dimensions. The results are presented with KDE models with (a) original 14-dimensions, (b) reduced 7-dimensions after PCA, and (c) reduced 5-dimensions after PCA.

every pixel is modeled as a GMM, which is learned through EM based approach. Experiments on real videos have proven the effectiveness of the proposed approach for local and global anomaly detection. Furthermore, by using the scene knowledge, we also show the improvements in object detection by using the feedback for the minimum object size and the background learning rate. This framework does not require explicit extraction of the main paths in the scene. This approach can easily benefit from online learning and can also be used for conventional applications like

predicting object path and scene exit points. In summary, the proposed framework is novel, robust, and can be generalized to more features than just motion and size.

In the second approach, we extend the first approach by modeling the distribution of motion patterns of object pairs. This is done through defining a composite random variable that combines transition vectors of two object concurrently present in the scene. The 14-dimensional probability density is learnt through KDE. The sparseness in higher dimensionality is handled through mean shift based sample refinement. Finally, Markov Chain is used to integrate the evidence over time. We present further improvement in the runtime by dimension reduction through PCA. We present encouraging performance on two different real scenes where we detect abnormal behavior like red light violation, illegal jaywalking, unusual person pickup, etc.

In the next two chapters we present our proposed models for human activities at the finer level in contrast to the models for coarse level presented here. The forthcoming chapters focus on the articulation of human body parts and utilize the trajectories of the body joints to model an activity. Chapter 4 describes the approach for recognizing activities of individuals and Chapter 5 presents the approach for predicting behavior of an individual.

CHAPTER 4: CHAOTIC INVARIANTS FOR HUMAN ACTIVITY RECOGNITION

We present a novel approach for classification of human activities in videos by using representative chaotic invariant features for each activity [4] . Human activities are modeled as nonlinear dynamical systems that are responsible to generate the observed time series data in videos. We utilize the trajectories of human body landmarks/joints (two hands, two feet, head and belly point) as the time series data. The observed data is then transformed to its respective higher dimensional state (*phase*) space through delay embedding. Dynamic and metric properties of the reconstructed phase space are used to determine the *chaotic invariants* including Lyapunov exponent, correlation integral, and correlation dimension. This set of features is then used to represent the original time series observed in the video. We prove the feasibility of our approach by recognizing human activities in standard video and motion capture data sets.

4.1 Introduction

Human activities consist of spatio-temporal patterns that are generated by a complex and time varying non-linear dynamical system. A complete description of this system will require enumeration of all independent variables, their interdependencies, differential equations controlling their evolution and a set of boundary conditions to be satisfied by the system. Ideally, one would like to have this complete description so that it can be used to control, predict, and extract features of the

dynamical system in a deterministic fashion. However, in practical scenarios obtaining a complete analytic description is extremely hard.

In computer vision literature, the problem of obtaining the description of a dynamical system is often overcome by selecting a set of variables defining the state space, and a function that maps the previous state to the next state. The type of the mapping function determines whether it is a linear, non-linear or stochastic dynamical system. For instance, human activities can be represented in terms of state variables defined as the image locations of body joints, followed by assuming that a linear [9], non-linear [86] or stochastic dynamical model [81] is controlling the evolution of these state variables. The unknown parameters of the dynamical model are learnt using a training data of human activities.

Our contention in this work is that by constraining the dynamical system to be of a particular type, one only *approximates* the true non-linear physics of human activities. In other words, by making assumptions about the type of the dynamical model, one tries to fit the experimental data to the model by finding values of the parameters that best explains the data. Rather than letting the data speak for itself about the type of the dynamical system, number of independent variables, degrees of freedom of the system, and values of unknown parameters. An analogous example of this type of approach from the field of probability theory is to assume the type of the probability distribution generating the data, say Gaussian, and then computing the mean and variance of the Gaussian. Rather than allowing the data to determine the actual shape of the probability distribution using kernel density estimation.

The aim of this approach is to derive a representation of the dynamical system generating the human activities directly from the experimental data. This is achieved by proposing a computational framework that uses concepts from theory of chaotic systems to model and analyze nonlinear dynamics of human activities, by using trajectories of body joints. There are few important points to note here: First, by proposing dynamical system generating human activities as a chaotic system, we are making the statement that there is a *determinism* present in the seemingly stochastic dynamics of human activities. This *determinism*, if exploited, can be used to derive richer features for activity recognition. Second, the proposed approach of modeling human activities directly from experimental data is superior to approximate modeling, since no assumptions have to be made about the type or form of the dynamical model.

Next, we present some of the relevant concepts from chaos theory that can be useful in understanding the forthcoming contents in this dissertation.

4.2 Chaos Theory Preliminaries

In this section we present the background material related to the theory of nonlinear dynamics and chaos. We believe that this quick overview will be helpful in understanding the rest of this dissertation. A dynamical system can be represented as a set of functions which describes how variables change in time. A dynamical system is termed nonlinear if the function defining the change in the system is nonlinear. A dynamical system may be stochastic or deterministic. In a stochastic dynamical system, new values are generated from a probability distribution, while in a deterministic dynamical system a single new value is associated with any current value.

Dynamical systems can be represented by state-space models, where state variables $X(t) = [x_1(t), x_2(t), ..., x_n(t)] \in R^n$ define the status of the system at a given time t. The state variables are often considered to be in subspaces of Euclidian spaces, but they more generally are in n-dimensional manifolds. The space of the state variables is often called the *phase space*. The state of the system evolves in accordance with a deterministic evolution function and the path traced by the systems states as they evolve over time is referred to as a *trajectory* or *orbit*. The collection of all trajectories from all possible starting points in the phase space of the dynamical system is called a *phase portrait*. An *attractor* is defined as the region of the phase space to which all the trajectories settle down to as time limit approaches infinity. If the attractor is not stable it is termed *strange*. The *invariants* of system's attractor are measures that quantify the properties that are invariant under smooth transformations of the phase space or control parameters. Invariants fall into three classes: 1) Metric 2) Dynamical and 3) Topological. Metric invariants include dimensions of different kind and multi-fractal scaling functions, while dynamical invariants include Lyapunov exponent. Topological invariants generally depend on the periodic orbits that exist in the strange attractor. *Embedding* is defined as a process of mapping one-dimensional signal to a d-dimensional signal.

Chaos theory is one of the ways to study nonlinear phenomena. The name 'Chaos Theory' comes from the fact that the systems the theory describes are apparently disordered, but theory is really about finding the underlying order in apparently random data. In other words, a chaotic system is a deterministic system which is globally stable, exhibit clear boundaries and displays sensitivity to the initial conditions. When applying chaos theory to a given a problem, the goal often is to extract information required to identify and classify strange attractors of the dynamical

system from the experimental data. The procedure can be broken down into a few relatively easy steps. These are: find a suitable embedding of the data, verify the existence of deterministic structure, compute dynamical, topological and metric invariants of the periodic orbits, and finally use the invariants for the identification purposes. The proposed framework for activity recognition is built around these basic steps. Intuitively speaking, for a computer vision practitioner chaos theory provides a way of determining the description of a dynamical system from a time series data. As long as one has the time series data, analysis steps described above can be applied. Few examples of the time series data that we come across in the field of computer vision would be trajectories, pixel intensity over time, flow vectors etc.

4.3 Framework

This section describes the algorithmic steps of the proposed activity recognition framework (see Figure 4.1). The main steps include:

1. Given a video of an exemplar activity, obtain trajectories of reference body joints, and break each trajectory into a time series by considering each data dimension separately.

2. Obtain chaotic structure of each time series by embedding it in a phase space of an appropriate dimension using the mutual information [41], and false nearest neighborhood algorithms [91].

3. Apply determinism test to verify the existence of deterministic structure in the reconstructed phase space.

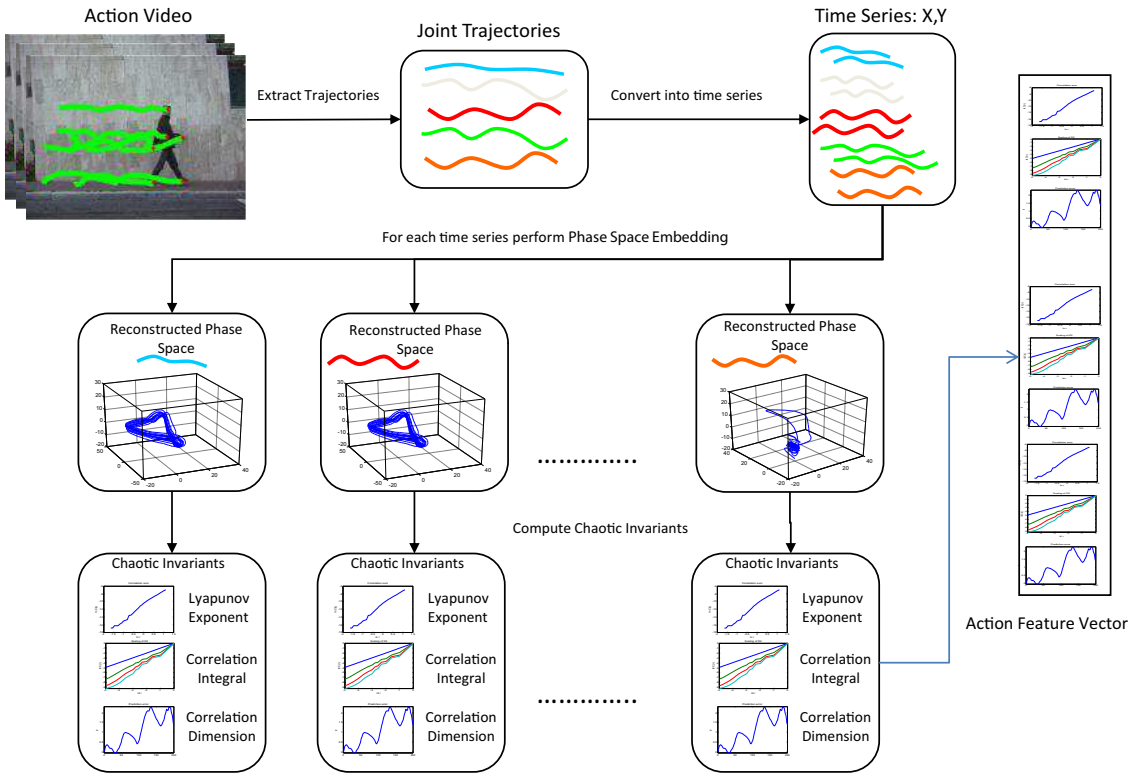


Figure 4.1: Overview of the chaotic invariant features extraction framework starting from an input video with tracked body joints (two feet, two hands, and the head).

4. Represent dynamical and metric structure of the reconstructed phase space in terms of the phase space invariants.

5. Generate global feature vector of exemplar activity by pooling invariants from all time series, and use it in a classification algorithm.

Next, these steps are explained in detail in the following subsections.

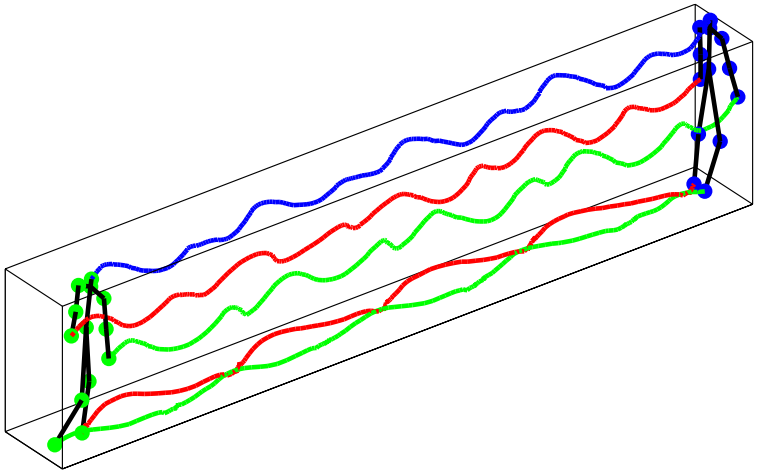


Figure 4.2: A sample set of 3-dimensional trajectories generated by head (blue), two hands (red & green) , and two feet (red & green) are shown for the running activity from the motion capture data set. The stick figure with green landmarks depict the first frame, and the one with blue landmarks represents the last frame.

4.3.1 Activity Representation

A trajectory corresponding to a body joint represents a deterministic nonlinear dynamical system. In our framework six body joints corresponding to two hands, two feet, head and belly are taken as the reference joints. To make the representation scale and translation invariant, trajectories of the first five joints are normalized with respect to the belly point. Hence, for any given activity we use five trajectories to represent the activity. We choose these reference joints as they provide sufficient information about most of the activities. Another consideration is that these joints are relatively easy to automatically detect and track in real videos, as opposed to the inner body joints which are more difficult to track. Figure 4.8 shows examples of set of trajectories for different activities in the case of real videos (2D trajectories), while Figure 4.2 shows trajectories for a running activity

from the motion capture data (3D trajectories). Note that, we are not solving the tracking problem in this section, therefore, we assume that the tracks are available to us. Formally, we represent the normalized trajectory corresponding to a joint b as a sequence of locations $Z^b = [\mathbf{z}_1^b, \mathbf{z}_2^b, ..., \mathbf{z}_t^b]$, where $\mathbf{z} \in R^k$ with $k = 2$ for image based measurement, and $k = 3$ for the motion capture data. Finally, we have $k \times N_B$ scalar time series for each exemplar activity, where N_B is the number of the reference joints.

4.3.2 Embedding

Embedding, as defined earlier, is a mapping from one dimensional space to a d-dimensional space. It is an important part of study of chaotic systems, as it allows us to study the systems for which the state space variables and the governing differential equations are unknown. The underlying idea of embedding is that all the variables of a dynamical system influence one another. Thus, every subsequent point, z_i^b, of a given one dimensional time series results from an intricate combination of the influences of all other system variables. Therefore, $z_{i+\tau}^b$ can be considered as a second substitute system variable which carries information about the influence of all other variables during time interval τ. Using this reasoning one can introduce a series of substitute variables $z_{i+2\tau}, ..., z_{i+d\tau}$, and thus obtain the whole m-dimensional phase space, where substitute variables carry the same information as the original variables of the system [84].

Formally, the embedding is achieved by using theorem of Takens [104], which states that *a map exists between the original state space and a reconstructed state space*. The theorem assures that one does not have to measure all the true state space variables of the system, as in fact almost

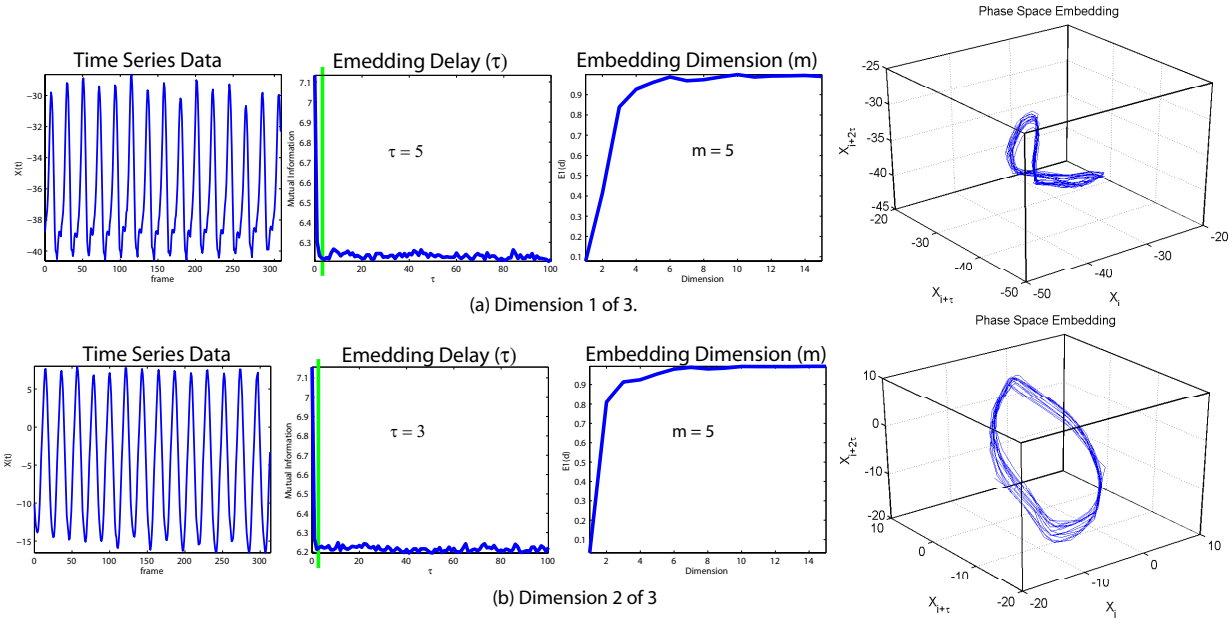


Figure 4.3: Depicts the embeddings of the time series corresponding to the right foot of the actor shown in Figure 4.2. The first column shows the time series corresponding to the x and y dimensions of the right-foot trajectory. The second column shows the plot of mutual information which is used to determine τ. The first minima value, marked by the green bar, reflects the optimal values of τ. The third column shows the plot of a measure $E1(d)$ [19], which can be derived from the false nearest neighbor algorithm, against different values of d. The value of d, after which the plot converges to a stable value, is chosen as the optimal embedding dimension. This happens to be at $m = 5$ in the current case. The fourth column shows the 3-dimensional projection of the reconstructed phase space for the chosen values of τ and d. This embedding is used to extract invariant features.

any one of the variables will be sufficient to reconstruct the dynamics. It also states that the dynamical properties of the system in the true state space are preserved under the embedding transformation. Thus, for a large enough embedding dimension d, the delay vectors $\mathbf{Y}^b(i) = [z_i^b, z_{i+\tau}^b, z_{i+2\tau}^b, ..., z_{i+(d-1)\tau}^b]$, generate a phase space that has exactly the same properties as that

formed by the original variables of the system. Over here, $z_i^b, z_{i+\tau}^b, z_{i+2\tau}^b, ..., z_{i+(d-1)\tau}^b$ represent scalar time series, belonging to one dimension of the trajectory, of the body joint b at times $t = idt$ to $t = (i + (d-1)\tau)dt$. Here, τ is known as the embedding delay. However, the embedding theorem does not provide a method to find the optimal values of τ and d. For estimating these values, we use the mutual information [41] and the false nearest neighbor algorithms [72]. In order to make this dissertation self-contained and readable, we are re-stating these algorithms from [84].

4.3.2.1 Estimating Embedding Delay

The estimation of delay parameter is based on the idea, that the mutual information between z_i^b and $z_{i+\tau}^b$ can be used to estimate a proper embedding delay τ. The algorithm considers two criterion: First, the value of τ should be large enough so that value of z^b at time $i+\tau$ is measuring something significantly different from the value of z^b at time i, and thus providing us with a new information which we do not have up till now. Second, the value of τ should not be larger than the time in which system loses memory of its initial state. The algorithmic steps are:

1. From the given time series $z_1^b, z_2^b, \ldots, z_t^b$, compute z_{min} and z_{max}.

2. Compute absolute value of their difference, $d = |z_{min} - z_{max}|$, and partition d into j equally sized intervals.

3. Compute:

   $I(\tau) = -\sum_{h=1}^{j}\sum_{k=1}^{j} P_{h,k}(\tau)\ln\frac{P_{h,k}(\tau)}{P_h(\tau)P_k(\tau)}$, where P_h and P_k denote the probabilities that the

variable assumes a value inside the hth and kth bin, and $P_{h,k}$ is the joint probability that z_i^b is in bin h and $z_{i+\tau}^b$ is in bin k.

4. Chose that τ as the embedding delay parameter for which $I(\tau)$ gives the first minima (Figure 4.3).

4.3.2.2 Estimating Embedding Dimension

For finding the optimal embedding dimension d we used the false nearest neighbor method proposed in [72]. The idea of the algorithm is to unfold the observed orbits from self overlap arising from the projection of an attractor of a dynamical system on a lower dimensional space. The algorithm makes use of the assumption that the phase space of a dynamical system folds and unfolds smoothly, and there are no sudden irregularities. This translates to the observation that if points are sufficiently close in a reconstructed phase space, then they should remain close during a forward iteration. If a phase space point has a neighbor that does not full fill this criteria then that point is said to have a false neighbor [84]. The steps for finding optimal d are:

1. Pick a point $p(i)$ in a d-dimensional space from the time series Z^b.

2. Find a neighbor $p(j)$ so that $\|p(i) - p(j)\| < \xi$.

3. Compute a normalized distance $R_i = \frac{|z_{i+d\tau}^b - z_{j+d\tau}^b|}{\|p(i)-p(j)\|}$, between $(d+1)$th coordinates of $p(i)$ and $p(j)$.

4. If R_i is larger then threshold R_{th}, then $p(i)$ is marked a having a false nearest neighbor.

5. Apply the equation in step 3 to entire time series for $m = 1, 2, \ldots$, until the fraction of points for which $R_i > R_{th}$is negligible.

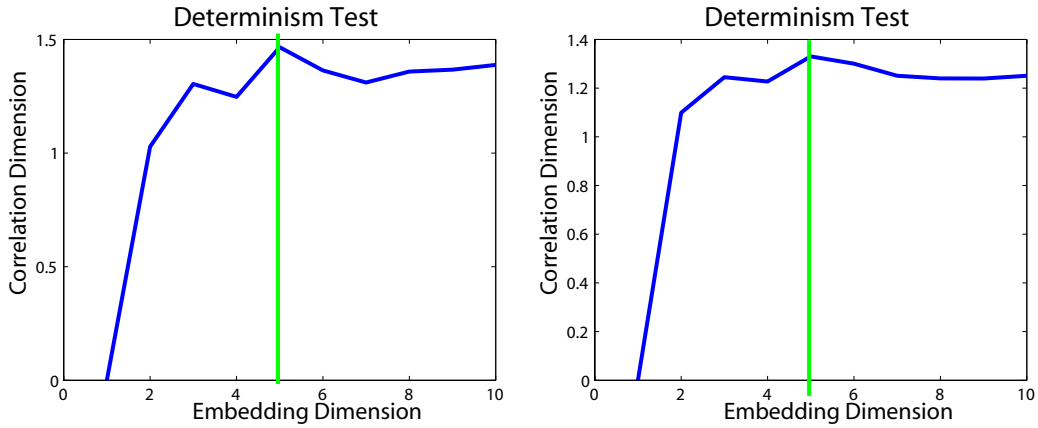
Figure 4.3 pictorially shows the process of finding optimal τ and d for two time series. It also displays 3-dimensional mapping of the reconstructed phase spaces. Once the values of τ and d are known, we slide a window of length d through the time series, and stack the d dimensional vectors row-wise into a matrix

$$X^b = \begin{pmatrix} z_0^b & z_\tau^b & . & . & z_{(d-1)\tau}^b \\ z_1^b & z_{1+\tau}^b & . & . & z_{1+(d-1)\tau}^b \\ z_2^b & z_{2+\tau}^b & . & . & z_{2+(d-1)\tau}^b \\ . & . & . & . & . \end{pmatrix}. \tag{4.1}$$

Note that each component of the d-dimensional vector is separated by an interval τ. Each row of the above matrix is now a point in the d-dimensional reconstructed phase space. We repeat the process for each time series, thus obtaining $k \times N_B$ reconstructed phase spaces for each activity.

4.3.3 Determinism Test

The purpose of this test is to get the evidence in support of our assertion, that there is a structure present in the trajectory data that can be exploited to obtain the representation of the underlying dynamics of human activities. It is performed on each of reconstructed phase space to distinguish irregular behavior resulting from deterministic chaos and the one appearing due to the noise. For this purpose, we employ a determinism test proposed in [110], where the idea is that neighboring trajectories in a small portion of the reconstructed phase space should all point in the same



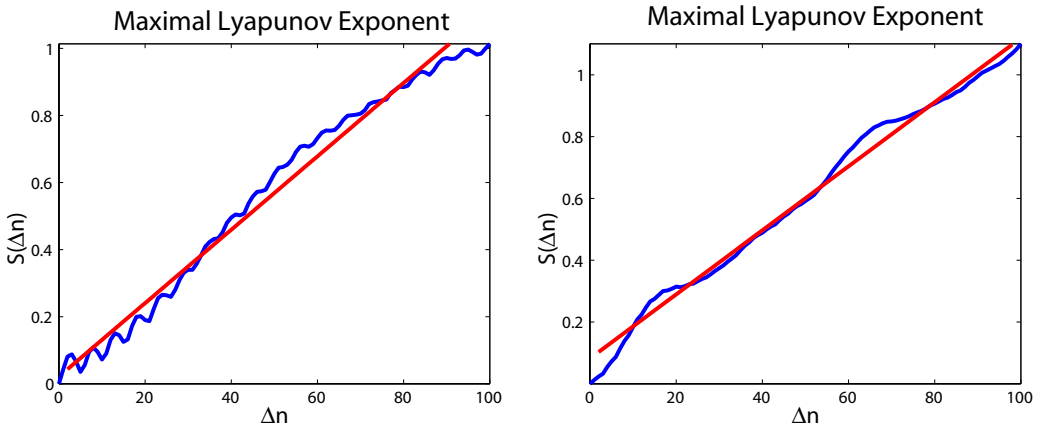
(a) Dim 1, d = 5 (b) Dim 2, d =5

Figure 4.4: The determinism test is performed by checking the convergence of the correlation dimension for the embedding dimension larger than m. In the case of a stochastic system, the value of correlation dimension (y-axis) increases monotonically with the increasing embedding dimension (x-axis). We show that the data under consideration indeed converges to the value of correlation dimension at the computed values of d (the green line) for the two time series shown in Figure 4.3.

direction, thus assure the uniqueness of solutions in the phase space which is a property of determinism. The outcome of this test (as shown in Figure 4.4) on our data validates the existence of determinism. That is, it reveals that the trajectories of the body joints indeed are generated by a deterministic process, and this justifies further analysis of the data by using the phase space invariants.

4.3.4 Invariant Features

Metric, dynamical and topological organization of orbits associated with a strange attractor of the reconstructed phase space can be used to distinguish different strange attractors representing



(a)Dim 1 (b) Dim 2

Figure 4.5: The computation of maximal Lyapunov exponent (for the right foot trajectory shown in Figure 4.2) from the plot of $S(\Delta n)$ against Δn. The slope of the line fitted to the curve provides a robust estimate of the maximal Lyapunov exponent. The estimated values here are 0.0104 for (a) and 0.0109 for (b).

different human activities. This organization is quantified in terms of phase space invariants. In this dissertation, we limit ourselves only to metric and dynamical invariants that include Maximal Lyapunov Exponent, Correlation Integral, and Correlation Dimension.

4.3.4.1 Maximal Lyapunov Exponent

Lyapunov exponent is a dynamical invariant of the attractor, and measures the exponential divergence of the nearby trajectories in the phase space. If the value of maximum Lyapunov exponent is greater than zero, that means the dynamics of underlying system are chaotic. In order to compute maximum Lyapunov exponent of reconstructed phase space, we employ algorithm given in [84]. The algorithm tests the exponential divergence of trajectories directly from the phase space trajectories.

To estimate the maximum divergence around a reference point $p(i)$ in the phase space, we start by finding all the neighbors $p(k)$ which are within distance ε. Here $p(i)$ is the ith row of the reconstructed phase space matrix X^b. The neighboring points are used as the starting point of nearby trajectories. The average distance of all the trajectories to the reference trajectory can be computed as a function of relative time Δn as follows:

$$D_i(\Delta n) = \frac{1}{r}\sum_{s=1}^{r} \mid z^b_{k+(d-1)\tau+\Delta n} - z^b_{i+(d-1)\tau+\Delta n} \mid, \tag{4.2}$$

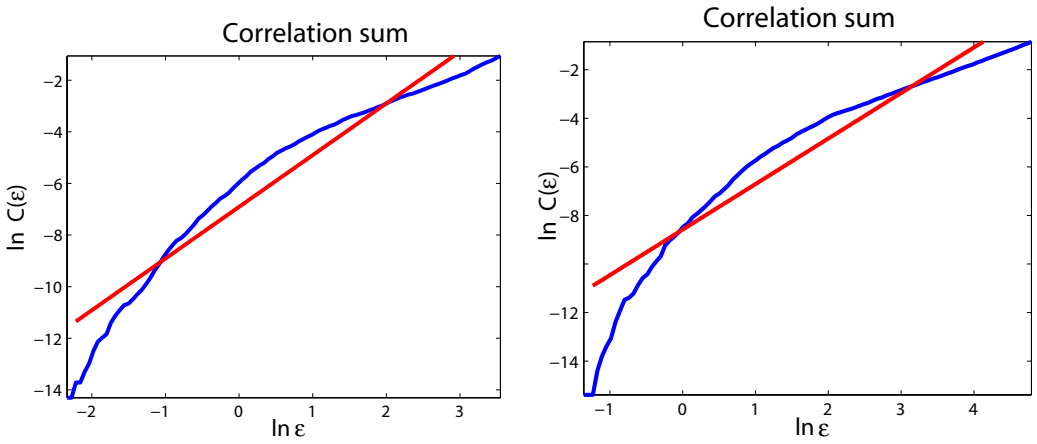
where s counts the different points $p(k)$, and there are total of r such points. Finally, the average of the logarithm of $D_i(\Delta n)$ is obtained for several reference points to get the effective expansion rate. That is we compute $S(\Delta n) = \frac{1}{c}\sum_{i=1}^{c} ln(D_i(\Delta n))$, where c is the number of reference points over which the process is repeated. Values of $S(\Delta n)$, computed for different Δn, and the maximum Lyapunov exponent is taken as the slope of the line fitted to the graph of $S(\Delta n)$ against Δn. Figure 4.5 shows this graph for the two time series shown in Figure 4.3.

4.3.4.2 Correlation Integral

The correlation integral is a metric invariant, which characterizes the metric structure of the attractor by quantifying the density of points in the phase space. It achieves this through a normalized count of pair of points lying within a radius ϵ. Formally, correlation integral $C(\epsilon)$ is defined as:

$$C(\epsilon) = \frac{2}{N(N-1)}\sum_{i=1}^{N}\sum_{j=i+1}^{N} \Theta(\epsilon - \parallel x_i - x_j \parallel), \tag{4.3}$$

where Θ is the Heaviside function. Note that, x_i in this case refers to a point in the phase space i.e. it corresponds to ith row vector of X^b. In our experiments, we computed $C(\epsilon)$ for a fixed values of



(a) Dim 1
Correlation Dim: 1.48

(b) Dim 2
Correlation Dim: 1.36

Figure 4.6: Computation of correlation dimension for the two time series shown in Figure 4.3. With increasing values of neighborhood radius ϵ (the horizontal axes), the values of the correlation integral (vertical axes) also increases. The slope of the line fitted to the curve provides an estimate of the correlation dimension.

ϵ and used it as a feature vector. Figure 4.6 shows the plot of the correlation integral for increasing values of ϵ.

4.3.4.3 Correlation Dimension

The correlation dimension also characterizes the metric structure of the attractor. It measures the change in the density of phase space with respect to the neighborhood radius ϵ. The correlation dimension can be computed from the correlation integral by exploiting the power law relationship $C(\epsilon) \approx \epsilon^d$, where d is the correlation dimension. The computation of the correlation dimension proceeds by plotting $C(\epsilon)$ and ϵ on a log-log graph. Again, the slope of the linen fitted to this graph provides a robust estimate of correlation dimension ,because the region in which power law

is obeyed appears as a straight line in the graph. Figure 4.6 shows this graph, along with the estimated values of the correlation dimensions for the two time series shown in Figure 4.3. The region whose slope is an estimate of the correlation dimension.

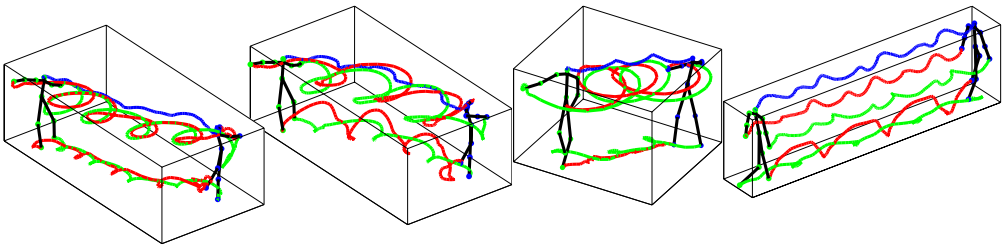
Another useful information about the activity can be obtained from the variance of the time series data, which we employ as a part of the feature vector in addition to the phase space invariants.

4.4 Experiments

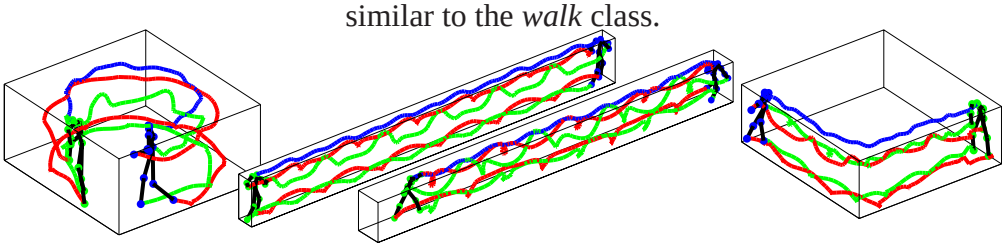
Experimental analysis is carried out on three data sets for human activity recognition. FutureLights data set [43] (see Figure 4.7), Weizmann data set [11] (see Figure 4.8), and UCF Sports Actions data set [101] (see Figure 4.10) are used to demonstrate the accuracy of the proposed approach.

4.4.1 FutureLights Motion Capture Data Set

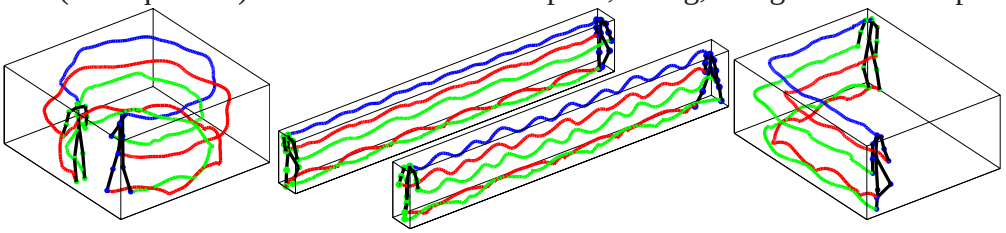
The first set of experiments was performed on the data set containing 3-dimensional motion capture sequences provided by FutureLight [43]. Figure 4.7 shows some typical sequences from this data set. In total, it contains 155 sequences of 5 activity classes, namely *dance*, *jump*, *run*, *sit*, and *walk* with $30, 14, 30, 33,$ and 48 instances, respectively. All five classes have significant intra-class variations. For example, the *run* class has variations in terms of speed (jog, run), stride length (short, long), bounce (low, high), and arm swing (low, high). The sequences in the run class, therefore, are created by several combinations of these parameters, and also include stopping and turning events. Similarly, the *walk* class contains these variations, in addition to a parameter for the pelvic swing (high, low). There are other variations like walking in a circle, turning around,



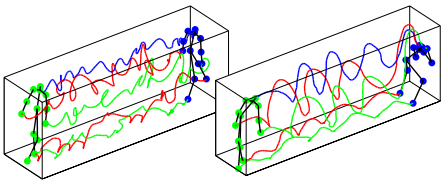
(a) *Dance* (30 sequences): includes a large variety of ballet sequences. A subset of these is very similar to the *walk* class.



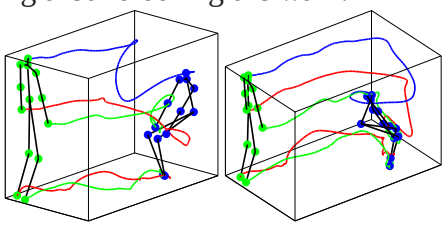
(b) *Run* (30 sequences): includes variation in speed, swing, and global motion pattern.



(c) *Walk* (48 sequences): represents the largest class with many variations of speed, swing, and events like stopping and turning around during the walk.



(d) *Jump* (14 sequences): mostly hopping while walking



(e) *Sit* (33 sequences): contains variations in sitting postures & directions

Figure 4.7: Sample sequences of few activity classes from the motion capture data set. The stick figures with green joints depicts the first frame of the sequence, while the stick figure with blue joints represent the last frame.

Table 4.1: Confusion table for the motion capture data set. We achieved mean classification accuracy of 89.7%.

	Dance	Jump	Run	Sit	Walk
Dance	28				2
Jump		13			1
Run	2	1	22	1	4
Sit				33	
Walk	3		2		43

stopping etc. The *dance* class contains stationary and moving ballet sequences, and some catwalk sequence, which in fact resembles closely to the *walk* sequences. The *jump* class contains jumping in place as well as jumping/hopping on one foot while walking. Finally, the *sit* class contains variations in the execution styles. In summary, all the activity classes contains significant intra-class variations. and therefore, this is a very challenging data set.

The initial input is in the form of trajectories of 13 body joints of the stick figure shown in Figure 4.2, but we only use 5 reference joints. We extract scalar time series from all five reference joints, resulting in 3 time series (x,y, & z) per reference joint and 15 time series per activity. Each time series is embedded separately using the procedure described in Section 4.3.2.2. A four dimensional feature vector is then constructed for each time series by computing Lyapunov exponent, correlation integral, correlation dimension and variance. After concatenation, for a given activity sequence this results in a 60-dimensional feature vector. For testing, we use the leave-one-out cross validation approach using the K-nearest neighbor classifier with $K = 5$.

The classification results achieved by this approach are shown in the Table 4.1. We achieved mean accuracy of 89.7% on the entire data set. Four *run* sequences were misclassified as the *walk*, which is understandable considering the similarity between these activities. Another main source of error was the confusion between the walking ballet sequences from the *dance* class and the *walk* class.

4.4.2 Weizmann Action Data Set

The second set of experiments was performed on Weizmann action data set [11], which depicts real actors performing different activities. Figure 4.8 shows examples of these activities. Specifically, the data set contains 81 videos with 9 different activities performed by 9 different actors. Given the data, the first step in the algorithm is the extraction of joint tracks for the six landmarks on the human body (two hands, two feet, the head, & the belly point). We used a semi-supervised joint detection and tracking approach for this experiment. That is, for computing trajectories for the reference joints, we extracted body skeletons and their endpoints using by using morphological operations on foreground silhouettes of the actor. Then an initial set of trajectories is generated by joining extracted joint locations using the spatial and motion similarity constraint. The broken trajectories and wrong associations were corrected manually. Note that the quality of the phase space embedding is dependent on the length of a time series, which implies that we need to observe the target activity for sufficiently long period of time (approximately 200 frames). However, the length of the videos in the data set varies from 27 to 80 frames. We overcame the problem by up-sampling and concatenating the original trajectories and thereby increasing the number of observations. Our

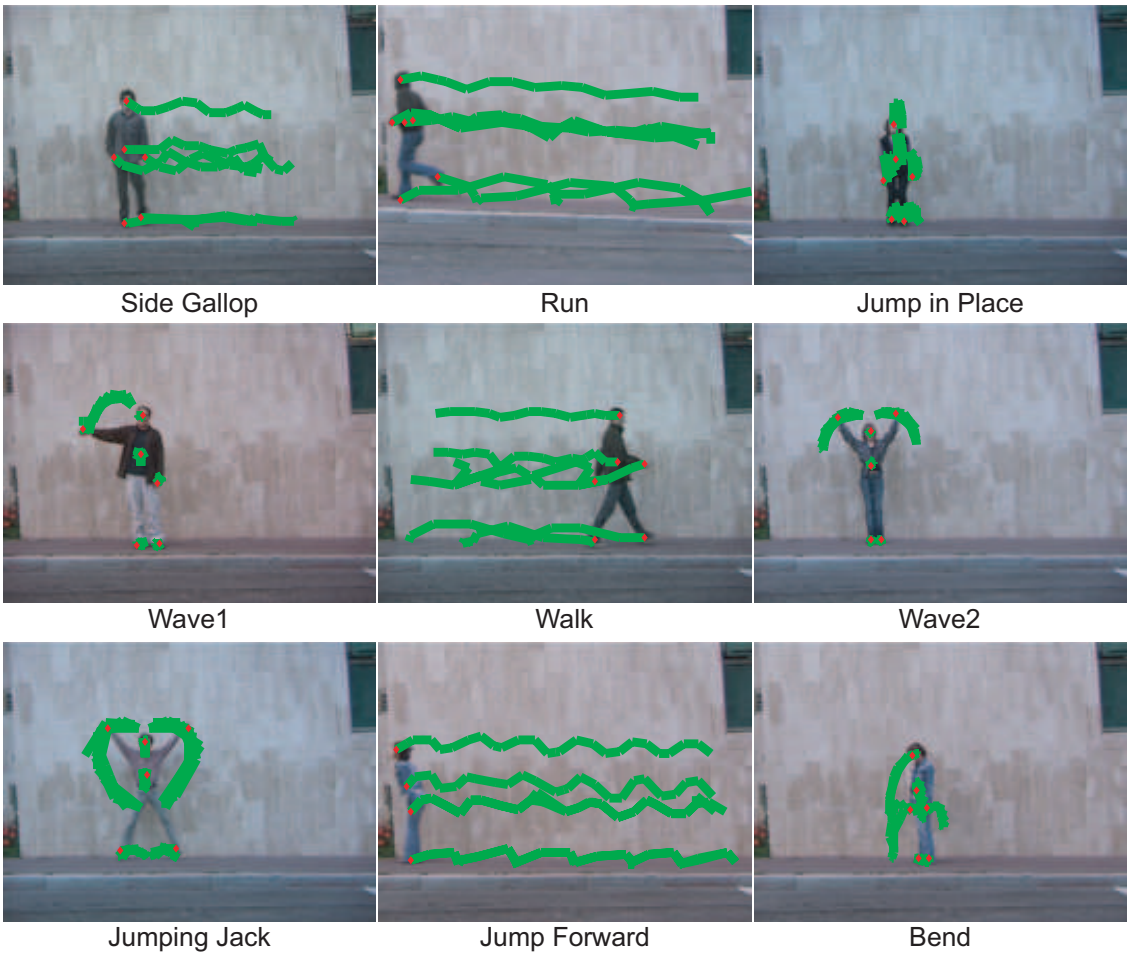


Figure 4.8: Nine different activities are used from the dataset provided by [11]. Trajectories from six landmarks (two hands, two feet, the head, and the body center) on human body are used as input to our method. These trajectories are used to extract invariant features of the reconstructed phase space that represent the underlying dynamical system.

experimental results have shown we are able to capture variations present in different activities by employing this approximation. Once the trajectories of five body joints relative to the centroid of foreground blob are recovered, we decomposed each of them into their two spatial components (x

Table 4.2: Confusion table for the Weizmann data set [11], where our algorithm has achieved mean accuracy of 92.6%.

	Bend	Jumping Jack	Jumping Forward	Jumping in Place	Run	Side Gallop	Walk	Wave1	Wave2
Bend	9								
Jumping Jack		9							
Jump Forward			5	2	2				
Jump in Place				9					
Run					8		1		
Side Gallop					1	8			
Walk							9		
Wave1								9	
Wave2									9

& y). This resulted in ten time series in total, which are then used to compute the invariants. After concatenating, for a given activity this resulted in a 40-dimensional feature vector.

The testing was performed performed by using leave-one-out cross validation. When using K-nearest neighbor, one sequence is kept as a test sequence while all the remaining sequences were used as training samples. We obtained a mean classification accuracy of 92.6% for all nine activities. The confusion table is shown in Table 4.2. It can be observed that only 6 out of a total of 81 videos were misclassified in these experiments. Two of the misclassified videos were from the *Jump Forward* activity, which were incorrectly labelled as *Run* activity. While two other videos were misclassified as *Jumping in Place*. The *Run* and *Side Gallop* activities have one misclassification each. The observation we would like to make over here is that these are isolated

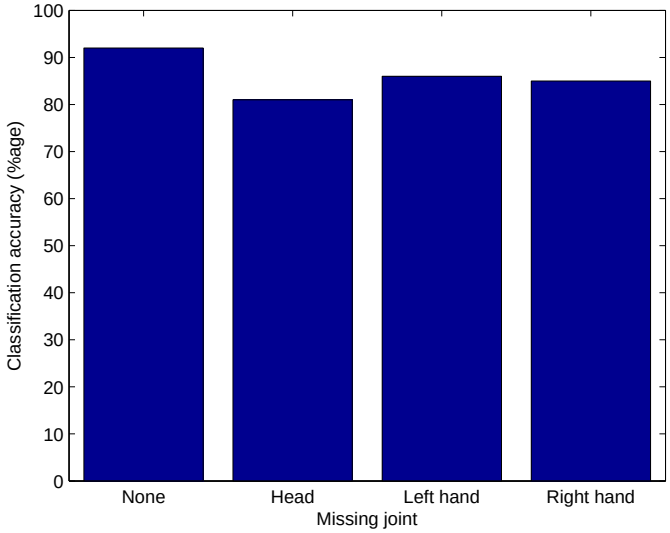


Figure 4.9: Comparison of classification accuracy is shown in several cases of missing joint trajectories. Head, right hand, and left hand are dropped one at a time from the Weizmann dataset.

errors, mostly for those activities which have quite a bit of similarity with each other, as is the case with confusing running with walking, or jumping forward with running.

In order to test the robustness of our method with respect to the number of available joint tracks, we performed a second set of experiments by selecting only a subset of the five reference joints. First, the head trajectory was removed from the set of joint trajectories used, and we achieved a mean recognition accuracy of 81.2%. In this experiment most of the errors were observed in bending and jumping activities. In the second experiment, we removed the left hand joint instead, which produced a mean recognition accuracy of 86.1%. Similar performance is achieved when only right hand trajectory is removed. The classification rates under these different scenarios have been summarized in Figure 4.9. We consider this as a satisfactory performance, as we were able

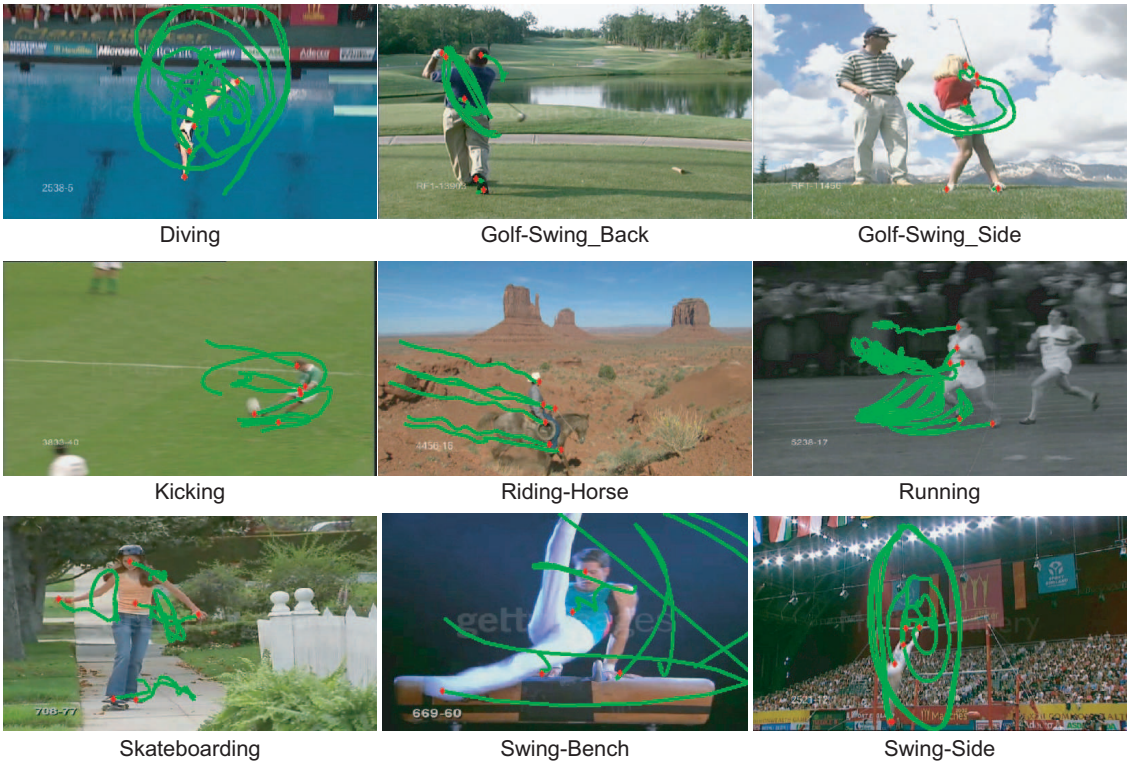


Figure 4.10: UCF data set was contains a set of actual sports activities captured from a moving camera. There are a total of 115 video sequences that were obtained from online video archives.

to maintain the activity recognition accuracy up to a reasonable degree even if one of the reference time series is missing. This shows that the proposed approach is not very sensitive to occlusion of individual body joints. At the same time, we observed that the classification accuracy for activities that are heavily dependent on the removed body joint (e.g. head in the case of bending) suffers more. But for activities like walking and running that involve multiple joints (two feet & two hands), removing one of these joints does not severely affect the overall classification accuracy.

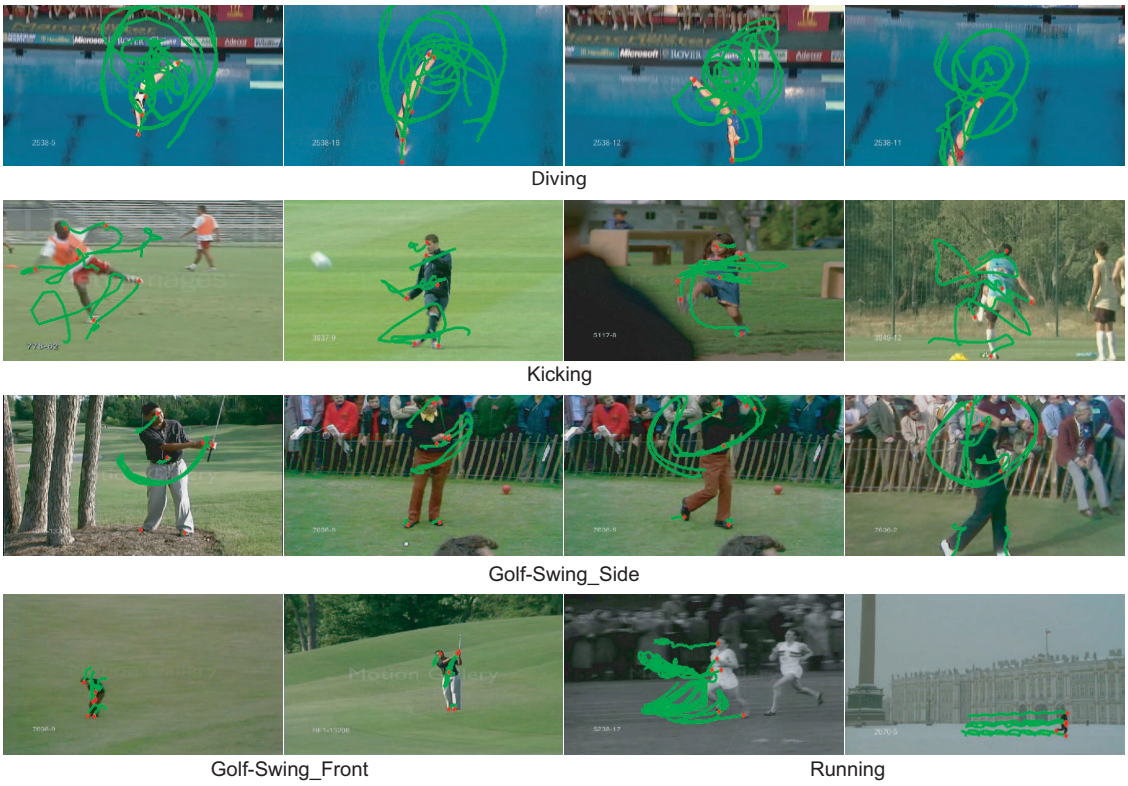


Figure 4.11: A small set of 16 sample videos is shown here for intra-class variations. The 6 joint trajectories used by our approach have been superimposed on each joint (highlighted by red point).

4.4.3 UCF Sports Actions Data Set

We have also experimented with a more challenging data set containing sports activities [101]. It contains a set of natural videos from actual sporting events. This includes activities like diving, golf swing, kicking, running, gymnastic swings, etc. Snapshot of activities in this data set are shown in Figure 4.10. The green trajectories overlaid in every frame show the six (head, left hand, right hand, belly point, left foot and right foot) input trajectories to our system. The sequences in this data set are captured from a moving camera and extracting the traditional foreground silhouettes

Table 4.3: Confusion table is shown for the UCF sports actions data set. Mean classification accuracy is 85.2%. The biggest confusion is between running and skateboarding actions, which can exhibit similar dynamics.

	Diving	Golf-Swing-Back	Golf-Swing-Front	Golf-Swing-Side	Kicking	Riding-Horse	Running	Skateboarding	Swing-Bench	Swing-Side
Diving	**13**									**1**
Golf-Swing-Back		4		1						
Golf-Swing-Front			8							
Golf-Swing-Side				4				1		
Kicking					20					
Riding-Horse			2	1		7		2		
Running					1	1	7	4		
Skateboarding				1		1		7		
Swing-Bench	1								15	
Swing-Side										13

was not feasible. In addition, the typical activities in this data set had exhibited self body occlusion. To concentrate on the analysis of the proposed approach, we manually obtained the input joint trajectories.

There are a total of 115 video sequences in this data set that were collected from several online video archives. These contain 14 diving, 5 golf swing (back), 8 golf swing (front), 5 golf swing (side), 20 kicking, 12 riding horse, 13 running, 9 skateboarding, 16 swing (bench), and 13 swing (side) sequences. Similar to the first two data sets the classification was performed using K-nearest neighbor classifier along with leave-one-out strategy. The mean classification accuracy on this data

set was found to be 85.2%. The confusion table is shown in Table 4.3. This classification rate is encouraging considering the complexity of the data set.

4.5 Summary

In this chapter we have presented a new approach for recognizing human activities when the finer level details of different body parts is available. The main contributions of this work include: a novel framework that characterizes the dynamics of human activities by using the theory of chaotic systems, a set of dynamical and metric invariant features of the strange attractor for classification, and a non-linear dynamical system based representation of human activities. An important result here is that we can represent an activity as a dynamical system for which we do not have an exact mathematical form. We have shown that the data-driven embedding and invariant features computed from it can be powerful for recognizing different dynamics. The mean classification accuracy on published data sets is comparable to the state of the art in this research area. Experimental validation of the feasibility and potential merits of carrying out activity recognition using this framework is demonstrated on various different scenarios. One limitation of the approach is the dependance on the joint trajectories of the human body. Tracking human body joints is outside the scope of this work. However, for this work, we adopt a a semi-supervised approach as explained previously. In the next chapter we present a new approach for tracking body parts in case of quasi-periodic actions.

In this chapter we have used the chaotic modeling of human activities for solving the recognition problem, however in the following chapter we address the problem of prediction. We propose

a multivariate extension of phase space embedding and show that a novel prediction approach is useful for human action synthesis and tracking.

CHAPTER 5: CHAOTIC MODELING FOR HUMAN ACTIVITY PREDICTION

We propose a new approach to model and predict time series data observed in different types of videos [7]. Such data would comprise of a sequence of observations over time, for instance, joint location or angle of a particular human body joint, pixel intensity at a particular location, etc. These time series would typically be generated by a deterministic nonlinear dynamical system with known initial condition. A good model of the underlying dynamics is important for predictions that are used in applications like video synthesis. When synthesizing longer sequences from a short sample video, it is desirable to generate realistic and smooth transitions. A trivial approach would be to concatenate the sample video multiple times, but this results in non-realistic transitions. Figure 5.1 shows an example of a scalar time series signal from running activity. This data is from one of the three dimensions corresponding to the 3D location of the human foot. The predicted signal (broken red) generated by the proposed approach creates a smooth transition and continues to depict the same dynamics as earlier. Such a mechanism could be useful in synthesizing repetitive human activities for long durations. This can have a variety of applications in computer vision and graphics including: human motion animation, occlusion handling, prediction for tracking, noise handling from motion capture data, dynamic texture synthesis, etc.

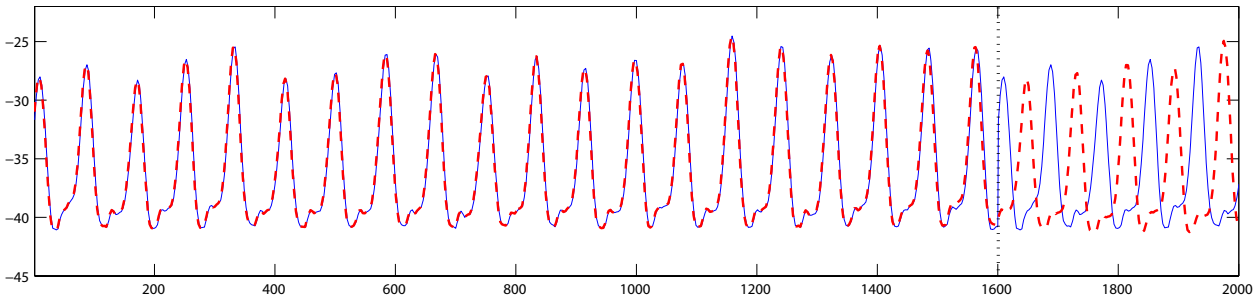


Figure 5.1: Abrupt vs. smooth transition: Original time series signal (solid blue) is repeated at the 1600 mark where it shows an abrupt transition. The predicted signal (broken red) shows a smooth transition and synthesizes the signal persistently.

5.1 Introduction

The observed scalar time series signals are transformed into a higher dimensional phase space through delay reconstruction (see Section 5.2.1). This results in a *strange attractor* which is characteristic of the underlying chaotic system. Note that a chaotic signal can be irregular and less predictable in the observed time series space, while in phase space it has a regular structure due to its deterministic nature. For prediction in phase space, several regression techniques can be used to compute the temporal mapping function. Many of these techniques often assume a particular underlying form of the mapping function (linear, polynomial, radial basis function etc.). However, in case of human activities we are not aware of the exact forms of the mapping functions responsible for generating the dynamics. Hence, instead of approximating a the functional form from the observed data, we rely on a more general approach. We use a nonparametric data driven model, based on kernel regression [80], to predict the future points along the strange attractor. These predictions are then transformed back into time series of longer duration with continuous motion. In

order to generate more realistic and synchronized multiple time series signals, we investigate the use of multivariate vs. univariate reconstruction for prediction. The use of multivariate time series embedding for human activities is novel. The predicted time series signals of body-pose parameters are used to synthesize and track human motion. In addition, the predicted pixel intensities are used to synthesize dynamic texture sequences.

The aim of this work is to investigate the relevant concepts from chaos theory and propose a novel and robust model for video synthesis. The novelty of this work lies in:

- The formulation of phase space reconstruction from the multivariate time series data of human activities. Previously (Chaptor 4), only univariate phase space models of human activities have been studied for activity recognition.
- A new deterministic dynamical model in contrast to previously popular stochastic noise-driven dynamical systems [36, 115].
- A new nonparametric model based on kernel regression in phase space.

We also provide experimental validation of viability of chaotic modeling approach for action synthesis as well as action tracking (see Section 5.3). This involves creating longer synthesized sequences of human activities using short sequences as a model. We have used standard motion capture data sets for this purpose. The comparison with few other synthesis approaches is also presented.

5.2 Proposed Approach

We investigate dynamical systems that define the time evolution of underlying dynamics in a phase (or state) space. First task is to find a way for phase space reconstruction from times series. The time series observations $\{x_0, x_1, \ldots, x_t, \ldots\}$ are transformed to the phase space vectors $\{\mathbf{z}_0, \mathbf{z}_1, \ldots, \mathbf{z}_t, \ldots\}$ through delay embedding, which is explained in Section 5.2.1. In the case of deterministic nonlinear dynamical (chaotic) systems, specifying a point in the phase space identifies the state of the system and vice versa. This implies that we can model the dynamics of a system by modeling the dynamics of the corresponding points in the phase space [60]. This idea forms the foundation of modeling the underlying chaotic system of unknown form and predicting future states. A system state is defined by a vector $\mathbf{z}_t \in \mathbb{R}^n$. The dynamics of these states are defined either by an n-dimensional mapping function

$$\mathbf{z}_{t+1} = \mathbf{F}(\mathbf{z}_t), \tag{5.1}$$

or by n first order differential equations. The latter approach is typically used for studying theoretical systems because the exact equations are rarely known for the experimental systems. The former approach, which is based on the mapping function, is more popular for the experimental systems. Section 5.2.2 describes a kernel regression based mapping function that we adopt for predicting future system states. These new states are transformed back to output time series as explained in Section 5.2.3. Figure 5.2 presents an overview of the steps involved in producing synthesized time series, starting with the model (training) time series that is the input to the process.

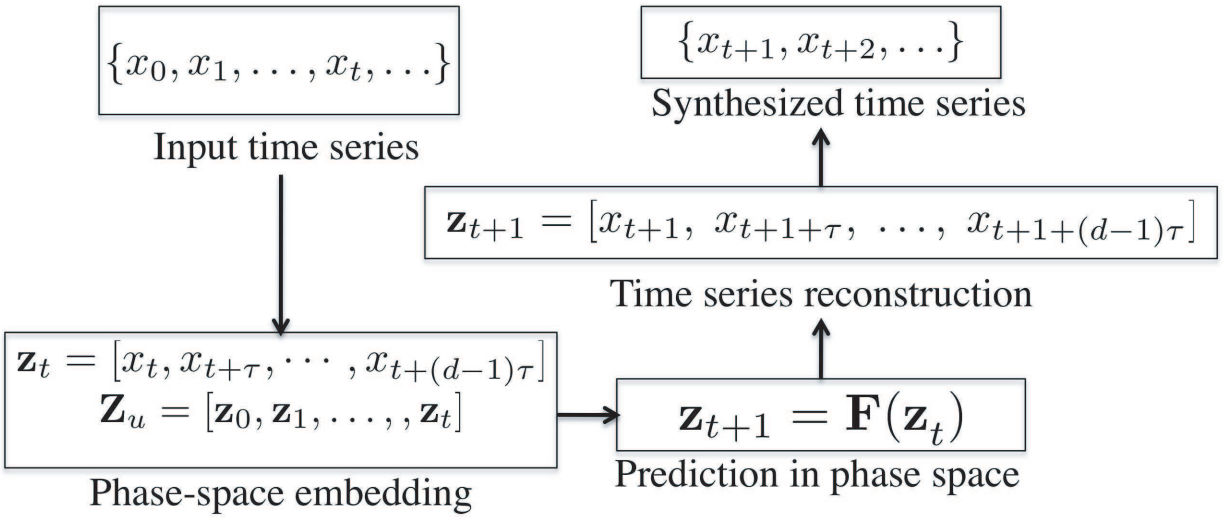


Figure 5.2: Main steps of the proposed approach for time series synthesis.

5.2.1 Phase Space Reconstruction

Phase space reconstruction is performed by the delay embedding of the observed data into phase space vectors. The details of the univariate delay embedding for human activities are provided in Section 4.3.2, however, we include relevant information for completion. Takens' delay embedding theorem forms the basis of this approach [104]. It states that *a map exists between the original state space and a reconstructed state space*. The theorem shows that the dynamical properties of the system from the true state space are preserved through the embedding transformation. Therefore, the delay vectors $\mathbf{z}_t = [x_t,\ x_{t+\tau},\ \ldots,\ x_{t+(d-1)\tau}] \in \mathbb{R}^d$, generate the phase space. The two parameters to be computed are lag τ and embedding dimension d.

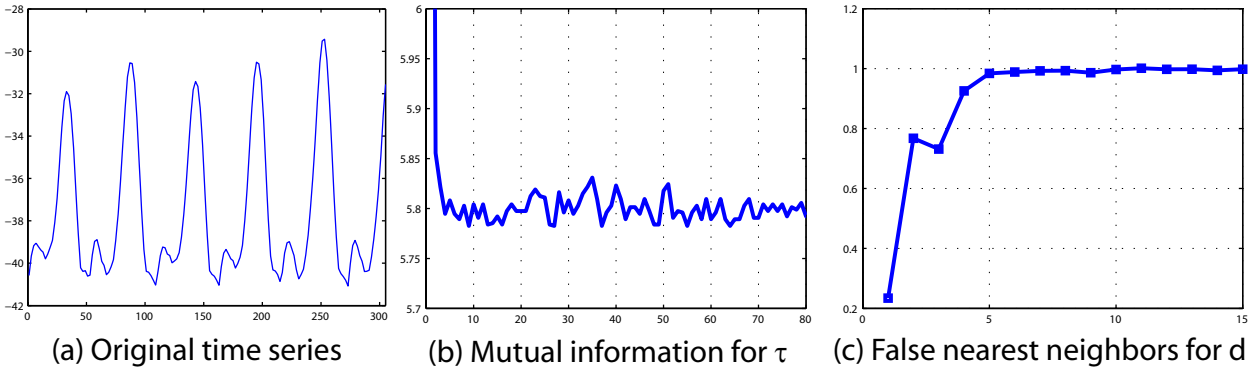


Figure 5.3: Steps for phase space reconstruction. (a) The observed univariate time series. (b) Mutual information plot to determine minimum delay (first local minimum, τ = 9). (c) The embedding dimension is computed by finding the smallest value that gives a small number of false nearest neighbors (converging to 1, d = 5).

The most popular approach for computing lag τ is based on the amount of mutual information between x_i and $x_{i+\tau}$ pair of observed values. The basic idea here is to look for the minimum τ for which the mutual information between observations is lowest. The details of the algorithm are available in [41]. Figure 5.3(a) shows a univariate time series generated by one dimension from the 3D location of the foot of a running person. Figure 5.3(b) shows the plot of possible τ values vs. amount of mutual information. The point of the first local minima of this plot is chosen as the lag τ. The optimal embedding dimension d can be computed by using the false nearest neighbors method proposed in [20]. The basic idea of this method is to find the smallest d, while minimizing the number of false nearest neighbors due to dimension reduction. Figure 5.3(c) shows the plot of possible values of d vs. fraction [0,1] of the points that do not have false nearest neighbors. Note that the fraction converges to 1 (100%) at $d = 5$, so choosing $d > 5$ would not be an optimal choice.

The values of τ and d are used to transform the univariate time series into the phase space (or delay) vectors $\mathbf{z}_t$ stacked as

$$\mathbf{Z}_u = \begin{pmatrix} \mathbf{z}_0 \\ \mathbf{z}_1 \\ \mathbf{z}_2 \\ \vdots \end{pmatrix} = \begin{pmatrix} x_0 & x_\tau & \cdots & x_{(d-1)\tau} \\ x_1 & x_{1+\tau} & \cdots & x_{1+(d-1)\tau} \\ x_2 & x_{2+\tau} & \cdots & x_{2+(d-1)\tau} \\ & & \vdots & \end{pmatrix}. \tag{5.2}$$

Note that each observed scalar value is repeated several time in this matrix. The sequence of the rows in this embedding matrix is important as it generates a trajectory in the phase space. Figure 5.4(a) shows the 3D projection of 5D phase space for the time series presented in Figure 5.3. This blue trajectory forms the *strange attractor* in the phase space. The metric, dynamical, and topological properties of this strange attractor are characteristic of the underlying nonlinear dynamical system [60]. We will be relying on modeling the evolution (flow) of the observed points along this strange attractor to predict the future locations.

This form of the embedding $\mathbf{Z}_u$ is feasible for prediction in the case of univariate time series. However, in computer vision we frequently observe time series generated by a dynamical system that involves multiple variables (dimensions) simultaneously. For instance, during human motion directly connected body joints impose certain constraints on the motion of each other. The trivial solution would be to proceed with performing univariate prediction separately for each dimension of the time series. We demonstrate through experiments that this approach breaks down due to the dependence between joint locations. Hence, a phase space reconstruction is desirable where prediction is performed for all the dimensions of a multivariate time series simultaneously.

Cao *et al.* [21] have shown that a simple yet powerful extension of the univariate embedding can be useful for the multivariate time series prediction. For a multivariate time series, with observations $\mathbf{x}_t = [x_{1,t},\ x_{2,t},\ \ldots,\ x_{D,t}]^T \in \mathbb{R}^D$, an appropriate phase space $\mathbf{Z}_m = [\mathbf{z}_0, \mathbf{z}_1, \mathbf{z}_2, \ldots]^T$ would be created by a set of delay vectors redefined as

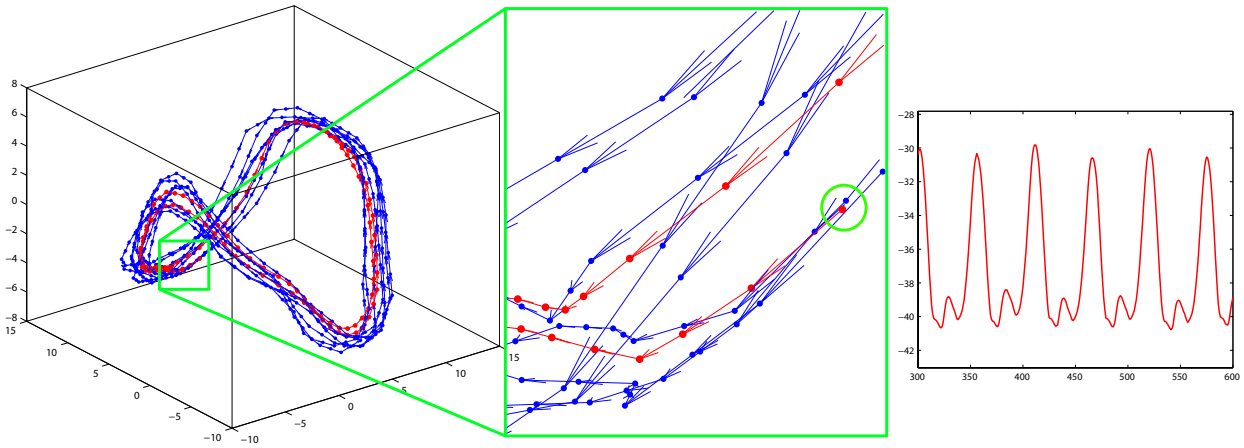
$$\begin{aligned}\mathbf{z}_t = [&x_{1,t},\ x_{1,t+\tau_1},\ \ldots,\ x_{1,t+(d_1-1)\tau_1},\\ &x_{2,t},\ x_{2,t+\tau_1},\ \ldots,\ x_{2,t+(d_2-1)\tau_2},\\ &\ldots,\\ &x_{D,t},\ x_{D,t+\tau_1},\ \ldots,\ x_{D,t+(d_D-1)\tau_D}] \in \mathbb{R}^{\sum_{i=1}^{D} d_i}.\end{aligned} \tag{5.3}$$

Here τ_i and d_i are respectively the delay and the embedding dimension for each one of the D dimension of time series. $\mathbf{z}_t$ maps to a point in the higher dimensional phase space and is linked to the next point $\mathbf{z}_{t+1}$ by the order in $\mathbf{Z}_m$ matrix. Figure 5.4(b) shows such points highlighted by dots and connected through arrows showing the direction of evolution.

5.2.2 Prediction in Phase Space

In order to perform prediction we need to compute the mapping function $\mathbf{F}$ (Equation 5.1). The exact form of $\mathbf{F}$ is unknown in case of general human motions. The "appropriate" selection of the model poses a challenge when one is not aware of the exact physics of the underlying dynamics. One popular form of the model is given by

$$\mathbf{z}_{t+1} = \mathbf{F}(\mathbf{z}_t) = \sum_{m=1}^{M} \mathbf{c}(m,t)\phi_m(\mathbf{z}_t), \tag{5.4}$$



(a) Phase-space with original time-series (blue) and predictions (red)

(b) Zoomed-in part with the initial condition for predictions (circled)

(c) Predictions transformed back to a scalar time-series

Figure 5.4: Predicting dynamics of a time series. Original time series is transformed into a strange attractor in the phase space. Kernel regression is used to estimate predicted values following behavior of neighbors. The predicted points in the phase space are transformed into a synthesized time series.

which is a linear combination of M possibly nonlinear functions ϕ_m with $\mathbf{c}(m,t)$ providing the coefficients. ϕ_m are usually chosen to be polynomials, radial basis functions, or logarithmic functions while the coefficient values $\mathbf{c}(m,t)$ are computed during functional approximation (e.g. least squares).

We avoid guessing a particular model by using a nonparametric model based on kernel regression [80]. The main idea is to estimate the mapping function using a weighted average of dynamics of neighboring points in the phase space. Hence, the mapping is given by

$$\mathbf{z}_{t+1} = \mathbf{F}(\mathbf{z}_t) = \sum_{k=1}^{N_n(\mathbf{z}_t)} (\mathbf{y}_{k+1} - \mathbf{y}_k + \mathbf{z}_t) w_k(\mathbf{z}_t, \mathbf{y}_k), \tag{5.5}$$

where $\mathbf{y}_k$ is one of the $N_n(\mathbf{z}_t)$ nearest neighbors of $\mathbf{z}_t$. Each of these neighbors has a corresponding next point $\mathbf{y}_{k+1}$ in the phase space. As shown in Figure 5.4(b), the vectors between the consecutive points are used in the neighborhood. The weights are computed from the kernel which is a decreasing function of distance from the reference point. Nadaraya-Watson [80] defined these weights as

$$w_k(\mathbf{z}_t, \mathbf{y}_k) = \frac{K_h(||\mathbf{z}_t - \mathbf{y}_k||)}{\sum_{k=1}^{N_n(\mathbf{z}_t)} K_h(||\mathbf{z}_t - \mathbf{y}_k||)}, \tag{5.6}$$

$$K_h(b) = \frac{1}{h} K\left(\frac{b}{h}\right), \tag{5.7}$$

where K is the kernel function which can be Guassian, Epanechnikov, etc, h is the bandwidth of the kernel and can be used for over smoothing. In our experiments we use $\mathcal{N}(0, 1)$ kernel and bandwidth $h = 0.5$. Such a chaotic modeling approach is generally: quite robust to noisy data, more accurate in experimental systems, and good for prediction while preserving important invariants of the dynamics [60]. Such an approach has the advantage of capturing a desirable balance between local and global parametric regression approaches. Local models are known to have the problem of large computational and memory requirements. On the other hand, the global models over generalize while computing one functional representation that models the whole attractor in the phase space.

Figure 5.4 shows the phase space reconstruction and predictions from the time series shown in Figure 5.3(a). The predictions are shown by red trajectories along with their directions of flow. Figure 5.4(b) shows the starting point (initial condition) of the prediction with closest neighboring points that contribute the most (through symmetric kernel) to the first prediction. Note that the

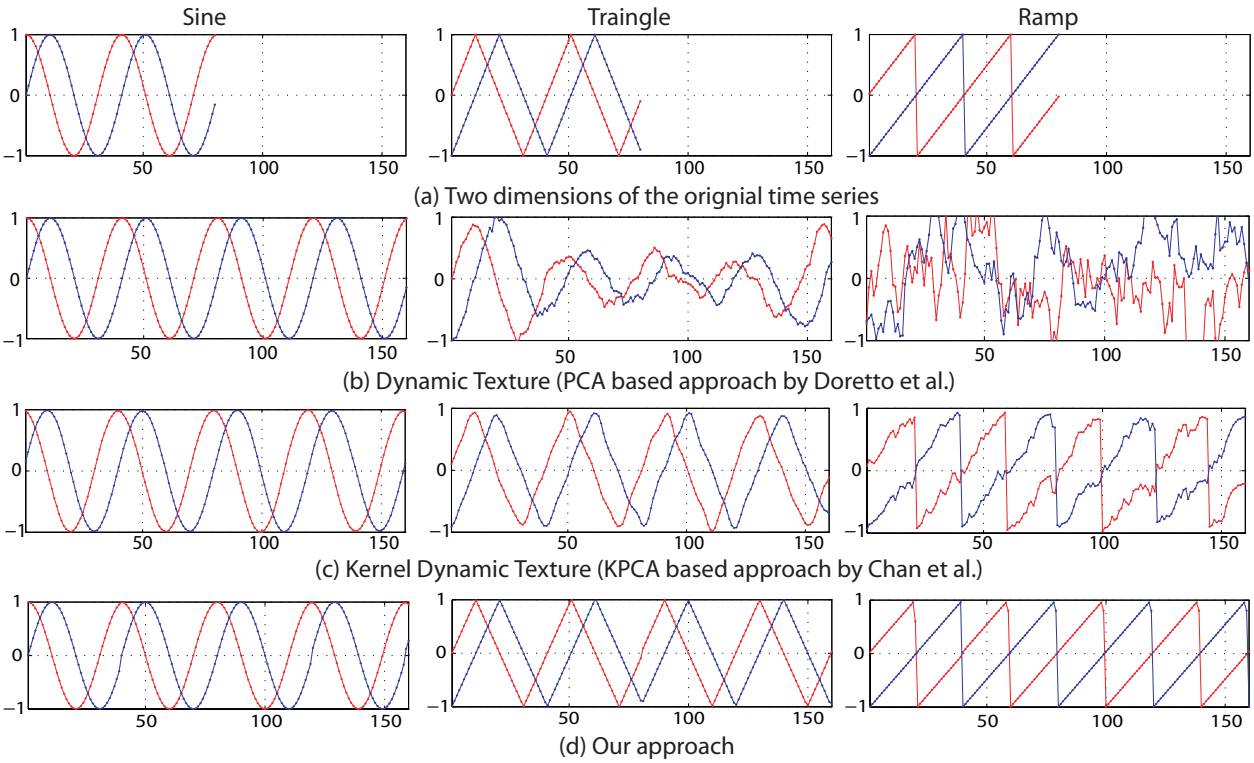


Figure 5.5: Comparison on synthetic data. (a) Sine, triangle, and ramp input time series. (b) and (c) show the synthesized output by Doretto *et al.*'s [36] and Chan *et al.*'s Kernel Dynamic Textures [25] respectively. (d) Synthesized output of our method provides more accurate reconstruction for all three signals.

first resultant arrow follows the immediate neighbors very closely. The predicted trajectory keeps evolving along the strange attractor following the system dynamics.

5.2.3 Time Series Reconstruction

To recover a time series from the predictions in the phase space we have to extract the time series from univariate $\mathbf{Z}_u$ or multivariate $\mathbf{Z}_m$ matrices. For the univariate case $\mathbf{Z}_u$ (see Equation 5.2) it is

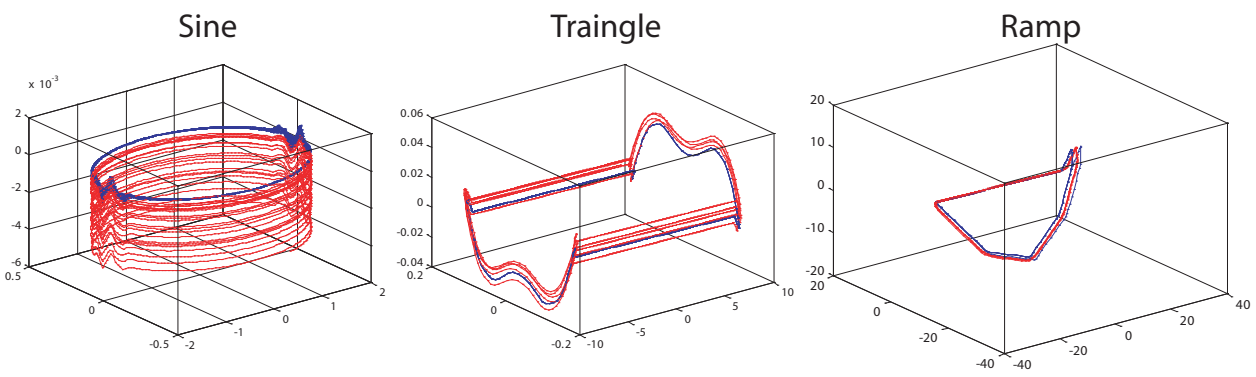


Figure 5.6: Visualization of the phase-space embeddings of the original signals (blue) as shown in Figure 5.5 and the corresponding predicted signal (red).

simply extracting the first column followed by last τ rows from the rest of the columns. For a Txd matrix $\mathbf{Z}_u$ this generates $T + (d-1)\tau$ time series observations

$$x_i \in \{\mathbf{Z}_u(1,i),\ \mathbf{Z}_u(k,T-j)\},$$

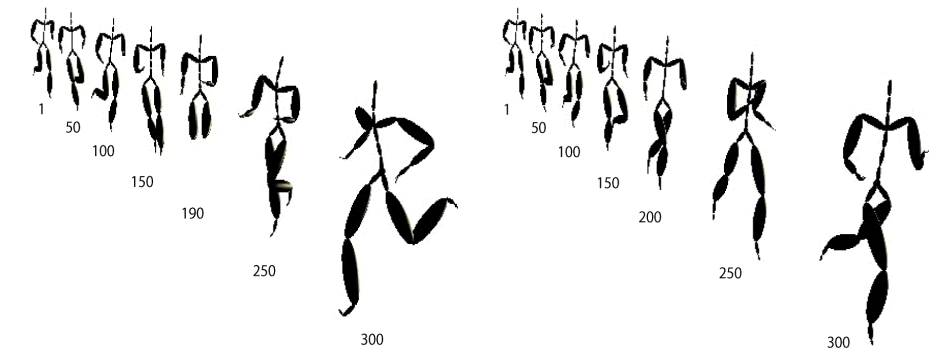
where $0 \leq i < T, \tau \geq j > 0, 1 \leq k < d$. In the multivariate case, $\mathbf{Z}_m$ matrix (see Equation 5.3) contains a row of D individual $\mathbf{Z}_u$ matrices. The multivariate time series is constructed by extracting D univariate time series from the corresponding $\mathbf{Z}_u$ as described above. Figure 5.4(c) shows an example of a univariate time series extracted from the predictions in the phase space shown in Figure 5.4(a). Figure 5.5 shows the output of time series synthesis on three synthetic signals where $D = 2$. The embedding parameters (τ, d) are calculated to be (4, 5), (3, 4) and (5, 7) for each dimension in sine, triangle and ramp signals respectively. It shows that the output of our approach is very similar to the source signal and is better than the two recent approaches used for dynamic texture modeling [36, 25].

5.3 Applications

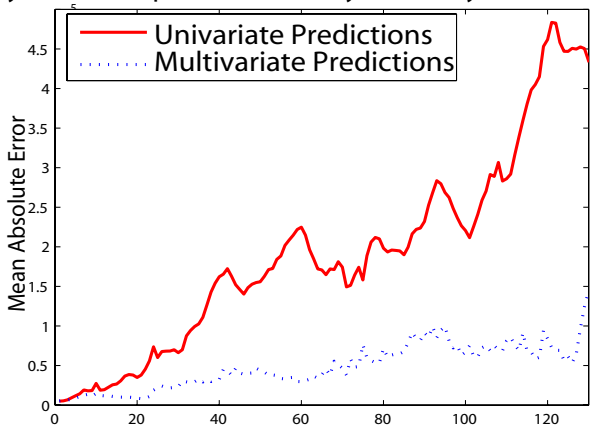
The proposed approach for predicting time series is applied to human activity synthesis and tracking. In addition, we also show that the presented model can be generalized to other types of motion, like dynamic texture. Several experiments were performed to evaluate the performance of our approach on published data sets and to compare the output with that of some of the well known methods in the literature.

5.3.1 Human Activity Synthesis

We use motion capture data to acquire source time series representing the position of the body landmarks during the activity. We use the motion capture data from FutureLight [43] and CMU [29] data sets for the human activity synthesis. Every frame in CMU and FutureLight sequences provides a 62 and 39-dimensional body-pose descriptors respectively. CMU's descriptor is composed of bone length and joint angles, while FutureLight is composed of the absolute 3D locations of the 13 body joints. A part of the sample sequence of the human activity is used to generate the observed time series $\mathbf{x}_t \in \mathbb{R}^P$, where P is the dimensionality of the body-pose descriptor. The multivariate phase space reconstruction produces $\mathbf{Z}_m$ embedding matrix for the sample activity. For a given starting point $\mathbf{x}_t$, the predictions and time series reconstruction is performed as explained before. This creates a sequence $\{\mathbf{x}_t, \mathbf{x}_{t+1}, \ldots\}$ of body-pose descriptors used for final video synthesis.



(a) Synthesis by univariate predictions (b) Synthesis by multivariate predictions



(c) Prediction error

Figure 5.7: Univariate vs. multivariate predictions for human motion. Univariate approach (a) shows irregular poses and its global transformations while multivariate approach (b) generates a smooth sequence with all valid poses. (c) Univariate predictions also result in a higher error than the multivariate predictions.

We have experimented with both univariate and multivariate predictions for this task. In the univariate case, each dimension of the pose descriptor is used independently to determine the phase space reconstruction followed by prediction. In the second case, multivariate prediction approach is used to evolve the predictions in an even higher dimensional phase space (order of P-dimensional). This provides the combined evolution of different dimensions of the pose descriptor.

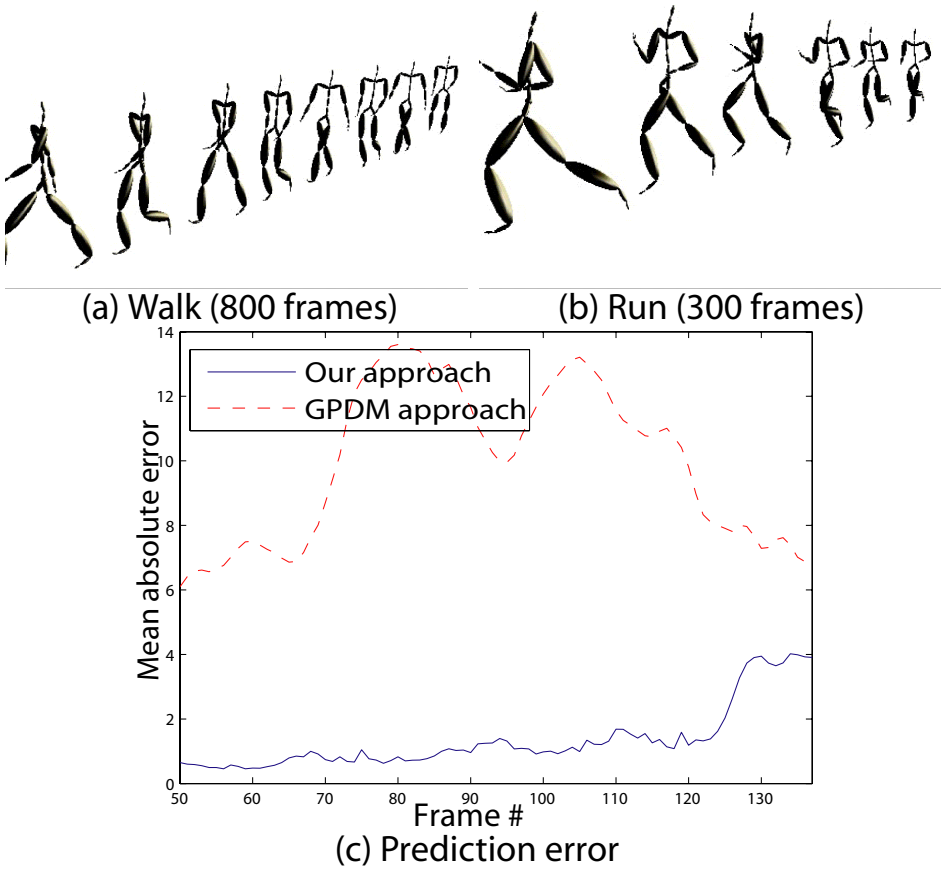


Figure 5.8: Human motion synthesis on CMU data set. Note that the difference between the walking and running body-poses is maintained after synthesis. (a) Every 100th frames is shown , (b) Every 50th frame is shown. (c) Quality of our predictions are compared against the ones generated by the GPDM based approach [40]. The ground truth between frame 50 and 137 is used to compute prediction error.

Figure 5.7 shows the keyframes from the same running sequence synthesized using the univariate (see Figure 5.7(a)) and the multivariate (see Figure 5.7(b)) predictions. These 300 frame long sequences have been synthesized from a 130 frames long model sequence. The keyframes in the multivariate case show normal body poses, however in the univariate case, strange poses are

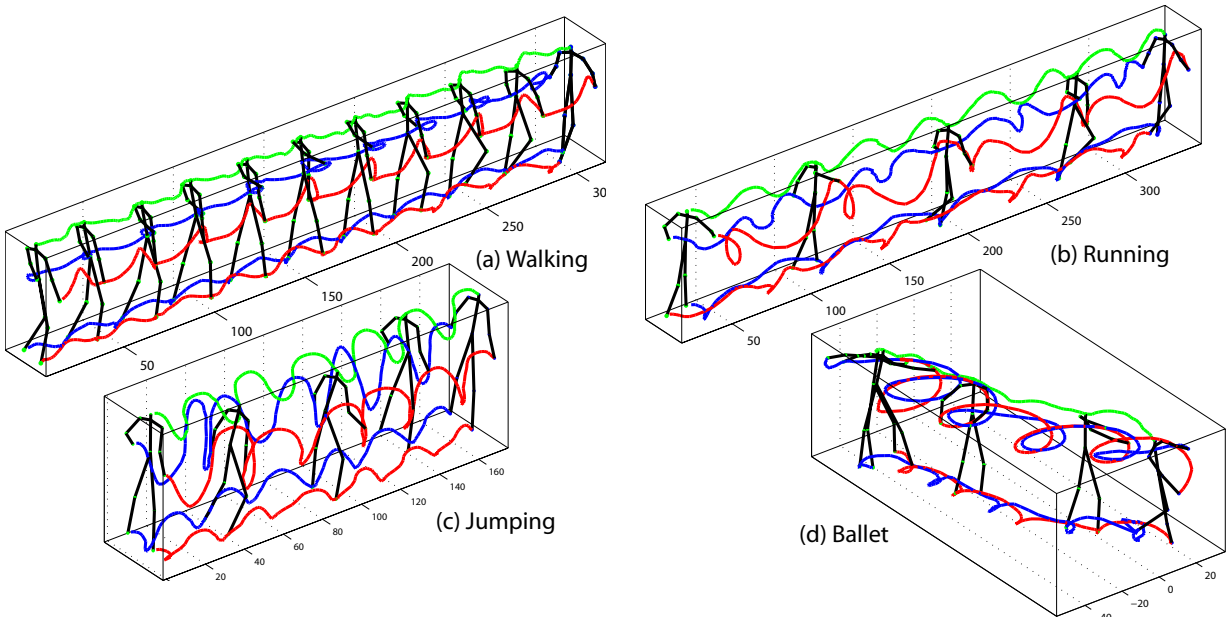


Figure 5.9: FutureLight data set. Synthesized sequences from each of the four different types of activities is shown. Here right hand & foot have red trajectories, left foot & hand have blue trajectories, while head has green trajectory. Faster speed in the running sequence (as compared to walking) can be noticed by the sparse stick figures that are drawn every 40 frames.

synthesized. Towards the end there is an unrealistic global rotation of the whole body. Figure 5.7(c) shows a graph of mean absolute error in the first 130 frames from both sequences that overlap with the model sequence. This clearly shows that the proposed multivariate formulation is critical for human activity synthesis.

Using the CMU data set, we show results on walking and running activities as shown in the Figure 5.8. The model sequences used in our experiments are typically 100 to 500 frames long. We synthesize sequences with up to three times the original length. The highest individual embedding dimension d_i observed during experiments was 7. We also compare the accuracy of predictions

with the output of GPDM based approach [40]. Figure 5.8 (c) shows a graph of mean absolute error in predictions by our approach (solid blue) and by Wang *et al.* [40]. The sequence (CMU $id : 09\_04$) shown in Figure 5.8 (b) is used for this experiment, where frame $1 - 100$ are used for creating the model and frame $50 - 137$ are used to compute the error in predictions.

Using the FutureLight data set, we synthesize walking, running, jumping, and ballet activities, as shown in Figure 5.9. We compute the relative locations of all other landmarks with respect to the belly (reference) point. This provides us with a 39-dimensional time series signal that will be predicted. The phase space embedding and predictions are computed through the aforementioned approach. During our experiments, the individual embedding dimension d_i would typically fall between 3 and 6 for these activities. The length of a typical model sequences used is between 220 and 500.

5.3.2 Human Activity Tracking

Prediction in a dynamical system has been shown to be useful for synthesis of periodic and deterministic motion. For the same kind of motion, predictions can also be useful for tracking the corresponding time series. In the case of human activity, the time series data corresponds to the location of body joints. Prediction of this time series can be useful for minimizing the search space in the tracking stage. We have proposed A similar, prediction based body parts tracking, approach has been previously [48]. In that case, geometric constraints were used to transform trajectories from a model video onto predictions in the test video. We show that the proposed tracking approach can be feasible in case of periodic activities. One limitation is that it requires the position

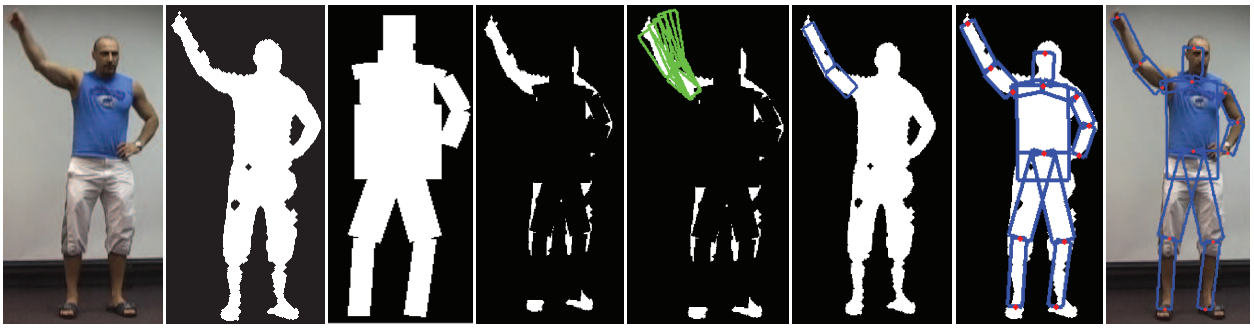


Figure 5.10: Steps involved in detection and tracking of human body parts through prediction: (a) Current source image, (b) output of background subtraction, (c) current state of the card-board body model for detecting right arm, (d) difference between images in (b) and (c), (e) set of predictions used, and (f) best match for right arm. Rest of the body parts are detected similarly, as shown on the foreground image (g) and source image (h).

of body joints to be known during the training phase. In some cases it can be obtained through a semi-supervised manner as explained in Chapter 4.

Figure 5.10 shows various stages of the detection and tracking approach adopted here and is similar to the one used in [48]. The main idea here is to utilize the predicted locations of a joints in a temporal window in order to find the best location in the current frame. This helps in significantly reducing the search space in the current frame. We start with the background subtraction assuming a stationary camera. A cardboard model is used to model the current pose of the body and is updated at every frame. As shown in the Figure 5.10(d), the part belonging to the left arm is isolated by subtracted the current pose estimate and the the background subtraction result. The exact location of the left arm is detected guided by a set of predicted locations/poses of the left arm. The best match if found by maximizing the overlapping are between the subtracted

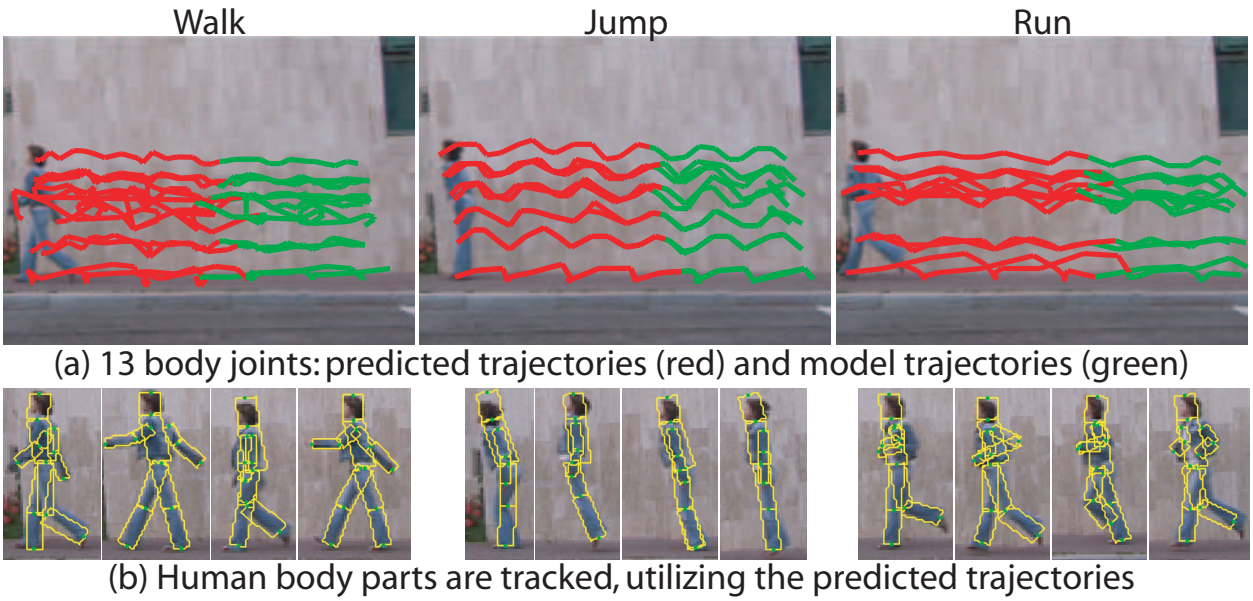
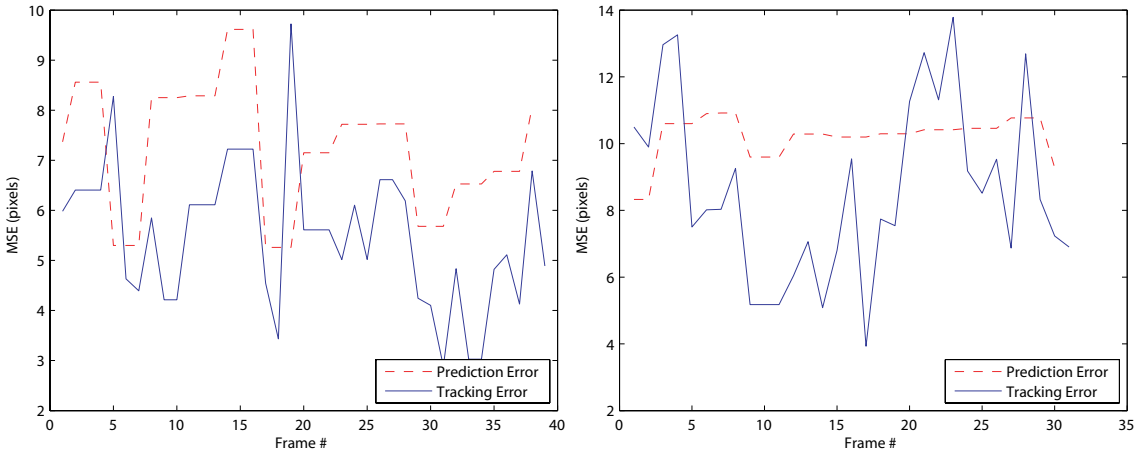


Figure 5.11: Predictions of body joint locations can be useful for tracking body parts in case of repetitive human actions.

foreground pixels and the predicted arm location. This process continues for the rest of the body parts and a complete body pose is generated.

We use Weizmann action data set for this task, where we demonstrate the results on three sample actions (walk, jump, & run) as shown in the Figure 5.11. We use the first half of these videos ($44, 36, \& 28$ frames respectively) as the model, where the 2D joint locations of the 13 body landmarks are available. The predictions for the 26-dimensional pose-descriptor are obtained in the same way as those for the synthesis task. As shown in the Figure 5.11, the predictions are very accurate when the motion is repeated regularly. To demonstrate the utility of these predictions for tracking body parts, similar to [48] , we use a card board body model along with the foreground silhouette feature. The overlap between the candidate locations (around predictions) of the body

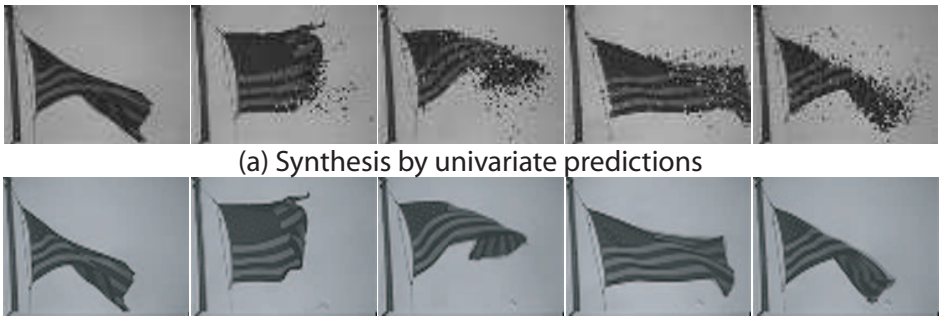


(a) Jumping (b) Running

Figure 5.12: Mean-squared error (MSE) is computed against the ground truth of the 13 joints in jumping and running videos. The tracking error is generally lower than the prediction error, as the initial estimates by predictions are refined after tracking.

part is maximized with the foreground silhouette to find the best estimate. We found that even with a simple tracking approach like this, we obtained encouraging tracking results. We observed some tracking artifacts in case of deviation from the style of action in the model video. We feel that our approach of generating predictions can provide useful prior for more involved tracking approaches.

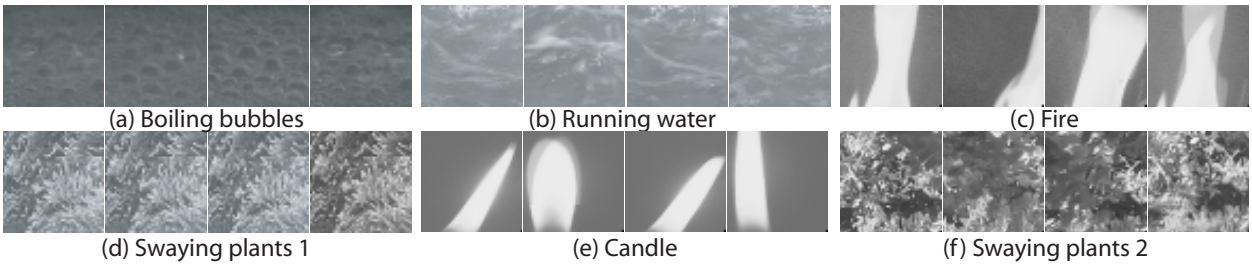
Figure 5.12 shows the quantitative comparisons of the mean-squared error in case of jumping and running actions. The error plot is computed at the predicted locations of the body joints and then at the final tracked locations of the body joints. Notice that the error reduces after tracking, when the predictions guide the search for local best matches for body parts. This shows that the proposed approach can be useful for tracking body joints in case of repetitive motion.



(a) Synthesis by univariate predictions

(b) Synthesis by multivariate predictions

Figure 5.13: Dynamic texture synthesis from Stripes video. (a) Predictions of many pixels quickly become unsynchronized from the neighbors causing the noisy pixels. (b) Multivariate predictions create more realistic and smoother videos.



(a) Boiling bubbles (b) Running water (c) Fire

(d) Swaying plants 1 (e) Candle (f) Swaying plants 2

Figure 5.14: Dynamic texture synthesis from UCLA data set. 75 frame long model videos are used to generate 225 synthesized frames.

5.3.3 Dynamic Texture Synthesis

We also demonstrate the synthesis of dynamic textures through the proposed approach of chaotic modeling. The dynamic textures have stochastic regularity in the spatial and temporal extent [85]. We investigate the determinism in the structure of dynamic textures through the proposed approach. The sequence of intensity values at each pixel is treated as a univariate time series, which is generated possibly by a chaotic system. We investigate the feasibility of both univariate and multivariate

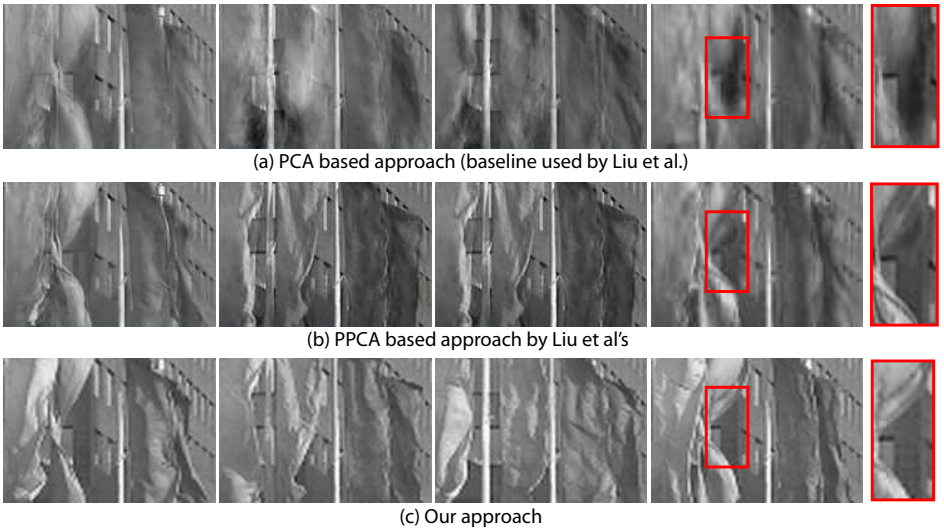


Figure 5.15: Dynamic texture synthesis from Flags video. We compare our method with the approach by Liu *et al.* [71] and the baseline method they used. Results obtained from our method are crisp and don't show ghost-like effects, as highlighted by the red box in the last column. Table 5.1 shows the prediction errors of these videos.

predictions in this case as well. The multivariate case is applied in small neighborhoods of 25×25 which creates 625-dimensional multivariate time series for each neighborhood. The actual dimensionality of the phase space would then be a sum of the individual 625 embedding dimensions d_i's. Figure 5.13(a) shows the synthesized video in the case of univariate predictions. Noisy pixels become more obvious as the video progresses because predictions diverge farther from ground truth. The multivariate case Figure 5.13(b) applies better spatial constraint and results in a synthesized video of better quality.

We first present synthesis results using the UCLA data set [94]. It contains 50 classes of different types of dynamic textures, including flames, trees, fountains, water etc. Each video contains

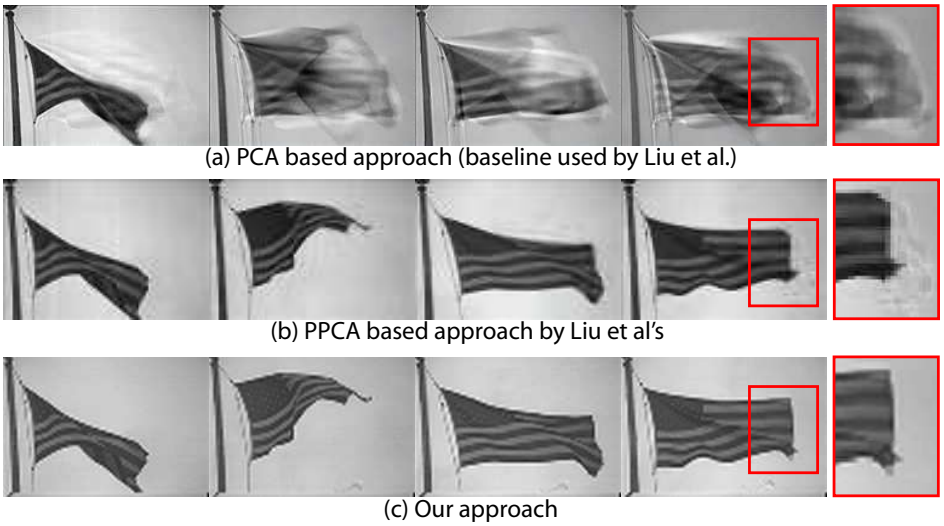


Figure 5.16: Dynamic texture synthesis from the Stripes video. We compare our method with the approach by Liu *et al.* [71] and the baseline method they used. Results obtained from our method are crisp and do not exhibit ghost-like effects, as highlighted by the red box in the last column.

75 frames of a cropped 48×48 textured area. Each pixel provides a scalar time series, whose embedding parameters are computed individually. This is followed by multivariate phase space reconstruction and prediction. The individual embedding dimension d_i for a pixel has been observed to lie between 4 and 9 for typical dynamics of the textures used here. Figure 5.14 shows a few of the synthesized frames from various types of videos in this data set.

A series of experiments have been performed to compare our approach to some of the popular approaches for dynamic texture synthesis. These include approaches by Chan *et al.* [25], Liu *et al.* [71], and Yuan *et al.* [115]. All of them provide means for quantitative and qualitative comparison with their approach, as well as the baseline PCA based linear dynamical system approaches and an improved version by Doretto *et al.* [36]. We performed experiments on the

Table 5.1: Mean squared error between the original and synthesized frames

Sequence name	Stripes (Figure 5.16)	Flags	River
PCA based approach (baseline in [71])	1119.8	1445.2	1198.0
PPCA based approach [71]	2117.9	579.5	551.4
Our approach	**12.2**	**17.8**	**8.6**

MIT dynamic textures data set [103], in order to present qualitative and quantitative comparison with these approaches. This data set contains videos that are typically 114×170 with 120 frames. These model videos were used to produce synthesized videos three times their length. The time series with pixel intensities is embedded into a higher dimensional phase space where prediction is performed. Figure 5.16 presents the output of our method, along with the corresponding output of the two approaches presented in [71]. The first is a baseline approach they used which relies on simple PCA with AR model. The second is their approach based on probabilistic PCA (PPCA). In Figure 5.16 we also highlight interesting area of the image with the red box. Note that both approaches in first two rows generate a ghost-like effect due to imperfect projection onto a few components, however, our approach preserves the quality. Table 5.1 presents quantitative comparison through mean squared error. This error is computed by the mean squared difference between the pixel values of the original and the predicted frames. We analyze the three videos (stripes, flags, and river) used in [71] and determine that our approach indeed outperforms both of these methods.

Table 5.2: Mean squared error between the original and synthesized frames

Sequence name	Fire (Figure 5.17)	Smoke-far	Smoke-near
Basic LDS (baseline in [115])	55264	230.7	402.6
Improved LDS [36]	55421	250.0	428.2
Closed-loop LDS [115]	1170	21.4	34.4
Our approach	**109**	**16.1**	**1.9**

Similarly, we perform another comparison with the closed-loop LDS by Yuan *et al.* [115], their baseline version LDS, and improved LDS by Doretto *et al.* [36]. Due to limited space, we only include the Fire sequence, which is the more challenging than the other two. The difference between the outputs of our approach and that from the first two approaches (basic and improved LDS) is obvious when looking at the figure. Table 5.2 clearly shows that our results are closer to the original video as compared to the out put of Yuan *et al.*

5.4 Summary

We have presented a new approach for time series prediction that can be used for fine level human activity modeling. We have presented application of these predictions for human activity synthesis, human body parts tracking and dynamic texture synthesis. In this chapter we have proposed a novel approach for modeling multivariate time series and performing prediction through a kernel regression based approach. We observed that multivariate phase space reconstruction is more suit-

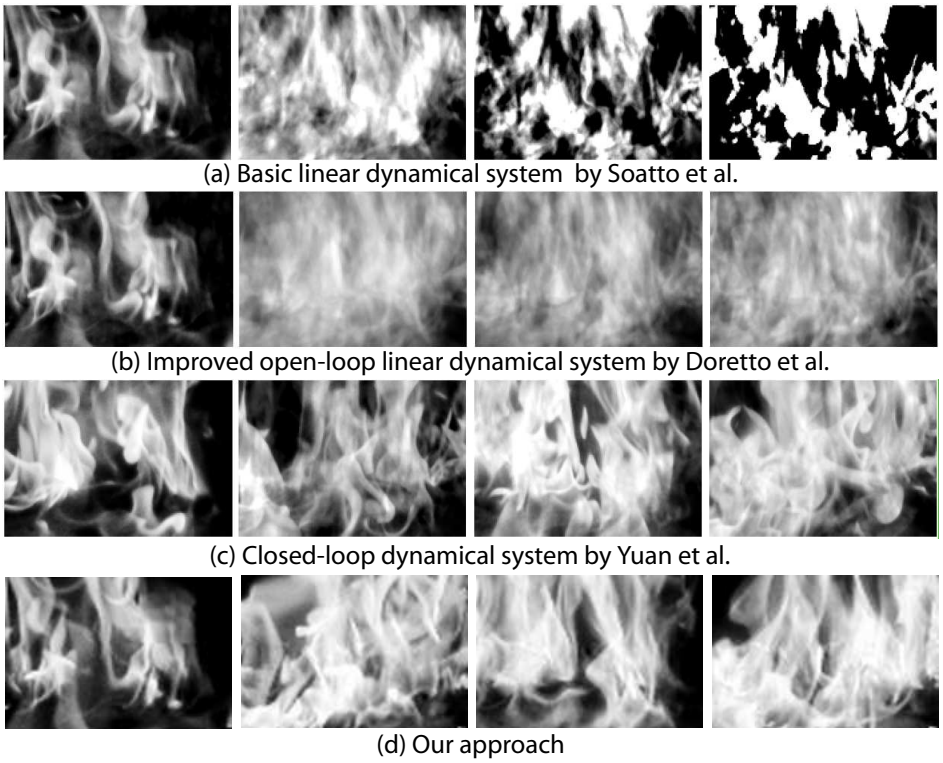


Figure 5.17: Dynamic texture synthesis from the Fire video. We compare our method with the that of Yuan *et al.* [115] and the baseline they used by Doretto *et al.* [36].

able for the task of prediction, as opposed to the univariate reconstruction used for recognition (see Chapter 4). We also show that the human activities can be modeled very well by a deterministic model that is inherently different from many noise-driven models. Noise dirven linear dynamical system has been a popular choice of various approaches in the past. We show the application of the proposed prediction approach to solve synthesis and tracking of human actions. Viability, robustness, and generalization of this model has been demonstrated empirically on standard data sets. Comparison with other approaches shows encouraging performance in terms of the quality of the synthesis.

We demonstrate the utilization of human body-pose predictions to address the problem of human body parts tracking. It is possible to use the proposed approach in case of repetitive human actions like walking, running, etc. Once the body joints have been tracked reliably throughout the test video, we can then apply the recognition approach (presented in Chapter 4) automatically. One limitation of the proposed tracking application is the requirement of the detected joint location in the training sequence. The use of semi-supervised method (as the one used in Chapter 4) can be one possibility to satisfy this requirement.

We conclude this dissertation in the next chapter with discussion and future directions.

CHAPTER 6: CONCLUSION

In this dissertation we have addressed modeling and recognition of human activities in videos. We have proposed a two pronged approach that provides different models for activities at the coarse and the fine levels. We also explained how the proposed work is aimed at filling the void in the literature. Our approach for anomalous activity detection is based on unsupervised learning, models motion of single objects as well as object pairs, avoids errors related to clustering tracks, and reuses the same scene model for improving object detection. For the activities at the fine level we proposed a strong model for modeling activities of individuals for recognition and prediction. We have presented a novel approach to model human activities as a dynamical system in the phase space. To the best of our knowledge, we have used the relevant concepts from chaos theory and non-linear dynamical systems for the first time to represent human activities and dynamic textures in computer vision literature. We have used a new set of features (chaotic invariants) for recognizing activities and proposed a new approach (kernel regression in phase space) for predicting human activities and dynamic textures.

Further discussion and future directions are discussed in Section 6.2. We briefly summarize the key contributions of this dissertation in the next section.

6.1 Summary of Contributions

The main contributions of this research to the literature include:

1. Scene modeling for unusual activity detection

   (a) Statistical scene model of single object parameters (motion and size) by using GMM pdf at every pixel location. Useful for real-time applications.

   (b) Statistical scene model of object pair parameters (concurrent motion and size) by using a global KDE pdf. Useful for detecting more complex anomalies.

   (c) Unsupervised learning and avoiding the errors related to clustering tracks into major paths in the scene.

   (d) The use of higher than first order velocities in modeling dynamics is useful to identify *global* anomalies in addition to the simpler *local* anomalies.

   (e) The proposed scene model is suitable to perform online learning of the evolving motion patterns in the scene.

   (f) Feedback of learnt scene model to the background subtraction module in order to improve object detection.

2. Chaotic invariants for human activity recognition

   (a) Investigation of the appropriateness of the theory of chaotic systems for human activity modeling and recognition.

(b) A non-linear dynamical system based representation of an action that without assumptions about the mathematical form.

(c) A new set of features to characterize nonlinear dynamics of human activities.

(d) Experimental validation of the feasibility and potential merits of carrying out activity recognition using methods from the theory of chaotic systems.

3. Chaotic modeling for human activity prediction

(a) Predicting dynamics without making any assumptions about the exact form (linear, polynomial, radial basis, etc.) of the mapping function.

(b) Multivariate phase space reconstruction for modeling human activities for prediction.

(c) A deterministic approach to model dynamics in contrast to the popular noise-driven approaches.

(d) Video synthesis and action tracking through kernel regression in the phase space.

6.2 Discussion and Future Directions

In this section we present our final comments and discussion on the three goals of this dissertation. In addition, we also present some possibilities for future directions to further this research.

6.2.1 Scene modeling for unusual activity detection

In order to address the first goal of modeling scenes and understanding activities at the coarse level, we have presented two novel approaches. The first approach models and learns the motion patterns

of individual objects in the scene, while the second one also models the interactions between objects pairs. While the first approach is more suitable for lightweight real-time applications, the second one is more powerful for detecting relatively more complicated behaviors in the scene.

In the first approach, we adopt an unsupervised learning approach that models object motion and size at every pixel location. The use of size feature in addition to velocity is merely for proof of concept. It provides means to classify objects based on their size. A more sophisticated feature for object classification can most certainly be used instead. The pdf of motion patterns at every pixel is modeled as a GMM, which is learnt through EM based learning approach. If the goal is real-time performance, one can reduce the spatial resolution and create a pdf for a local (e.g. 5×5 pixels) neighborhood instead of every pixel. Another benefit of using GMM is the convenience of making the model adaptable to the changing situation in the scene. We have not presented results in this dissertation with the adaptability but it can be incorporated in the future if required. The GMM parameters will be updated in an online fashion when the new observations become available, similar to background modeling [99].

In the second approach, we extend the statistical model by modeling the distribution of motion patterns of object pairs. This is done through defining a composite random variable that combines transition vectors of two object concurrently present in the scene. The 14-dimensional probability density is learnt through KDE. The sparseness in higher dimensionality is handled through mean shift based sample refinement. Finally, Markov Chain is used to integrate the evidence over time. We present further improvement in the runtime by dimension reduction through PCA.

We have shown improvements in object detection by using learnt scene model feedback for the minimum object size and the background learning rate. In addition to the object detection module, one can also improve the performance of the multi-object tracking module by incorporating the scene knowledge. For instance, if a Kalman filter [109] based model is being used, one can make measurement-noise and process-noise dependent on the most probable velocity in each region. The idea would be to use larger measurement noise, for instance, in case of faster motion of vehicles and vice versa. Other useful applications of the learnt scene model can include prediction of object path, occlusion handling while tracking, and finding source/sink (i.e. entry/exit) points in the scene.

6.2.2 Chaotic invariants for human activity recognition

We have presented a new approach for recognizing human activities when the finer level details of different body parts is available. Previously we have seen the use of dynamical systems as a way to model human actions. The approach presented here avoids the assumption of the linear model or specific form (polynomial, radial basis function, etc.) of non-linearity. An important result here is that we can represent an activity as a dynamical system for which we do not have an exact mathematical form. We have shown that the data-driven embedding and invariant features computed from it can be powerful for recognizing different dynamics.

We have used only three types of features from metric and dynamical groups of invariants. There are other features/measures that also fall into the same groups and could be evaluated. In addition, there is another group of invariant features that contains topological invariants. These features capture the physical topology of the embedded strange attractor and could be used to

for discrimination purpose. We experimented with two topological features: linking numbers and relative rotation rate. The limitation of these features was their validity only in low (less than three) dimensional phase spaces [46]. In our experiments with human activities, the dimensionality of the phase space was typically higher than three which made the utility of topological features very limited.

One limitation of the approach is the dependence on the joint trajectories of the human body. For this work, we adopted a semi-supervised approach as explained previously. There are several other approaches that can be useful for estimating human body pose, especially when the camera is stationary. In case of quasi-periodic actions we were able to obtain good results for body parts tracking through the proposed prediction based approach presented in Section 5.3.2. In order to avoid the dependence on these body joint trajectories, one possible direction of future work is to explore other types of features that can be extracted from images. Such a feature can be a shape descriptor of the body, so that a series of these features (used as a multivariate time series) can be used to express different types of actions. Possible candidates could be shape context [77] or histograms of gradient direction [32].

6.2.3 Chaotic modeling for human activity prediction

We extend the chaotic modeling of human activities for solving the prediction of moving body parts. We have presented application of these predictions for human activity synthesis in motion capture data and human body parts tracking in videos. In addition, we also show that the proposed model can be generalized to other types of dynamics, i.e. dynamic textures. In this chapter we have

proposed a novel approach for modeling multivariate time series and performing prediction through a kernel regression based approach. We observed that the multivariate phase space reconstruction is more suitable for the task of prediction, as opposed to the univariate reconstruction used for recognition (see Chapter 4). The multivariate reconstruction results in high dimensionality of the phase space, which is not a significant problem when the goal is to perform kernel regression. On the other hand, the high dimensionality could be a problem in other applications depending on the kind of features to be computed from the embedded strange attractor. The metric invariants, for instance, depend on the density of the points in the phase space and the quality of features is expected to deteriorate as the dimensionality increases. Another type of chaotic features is topological, that have been shown to hold only up to three-dimensional phase spaces [46]. Hence, the choice of multivariate phase space reconstruction should be made depending on the type of information or features to be derived from the strange attractor.

We demonstrate the utilization of human body-pose predictions to address the problem of human body parts tracking. It is possible to use the proposed approach in case of quasi-periodic human actions like walking, running, etc. Once the body joints have been tracked reliably throughout the test video, we can then apply the recognition approach (presented in Chapter 4) automatically. One limitation of the proposed tracking application is the requirement of the detected joint location in the training sequence. The use of semi-supervised method (as the one used in Chapter 4) can be used for this purpose.

LIST OF REFERENCES

[1] J. K. Aggarwal and Q. Cai. Human motion analysis: A review. *Computer Vision and Image Understanding*, 1999.

[2] J. K. Aggarwal, Q. Cai, W. Liao, and B. Sabata. Articulated and elastic non-rigid motion: A review. *Workshop on Motion of Non-Rigid and Articulated Objects*, 1994.

[3] K. Akita. Image sequence analysis of real world human motion. *Pattern Recognition*, 1984.

[4] S. Ali, A. Basharat, and M. Shah. Chaotic invariants for human action recognition. *ICCV*, 2007.

[5] S. Ali and M. Shah. A lagrangian particle dynamics approach for crowd flow segmentation and stability analysis. In *CVPR*, 2007.

[6] A. Basharat, A. Gritai, and M. Shah. Learning object motion patterns for anomaly detection and improved object detection. *CVPR*, 2008.

[7] A. Basharat and M. Shah. Time series prediction by chaotic modeling of nonlinear dynamical systems. *ICCV*, September 2009.

[8] A. Basharat, Y. Zhai, and M. Shah. Content based video matching using spatiotemporal volumes. *Comput. Vision Image Understanding*, 110(3):360–377, 2008.

[9] A. Bissacco, A. Chiuso, Y. Ma, and S. Soatto. Recognition of human gaits. *CVPR*, 2001.

[10] M. J. Black, Y. Yacoob, A. D. Jepson, and D. Fleet. Learning parameterized models of image motion. *CVPR*, 1997.

[11] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *ICCV*, 2005.

[12] A. Bobick and J. Davis. An appearance-based representation of action. *ICPR*, 1996.

[13] A. Bobick and J. Davis. Real-time recognition of activity using temporal templates. *IEEE Workshop on Applications of Computer Vision (WAVC)*, 1996.

[14] A. F. Bobick and A. D. Wilson. A state-based technique for the summarization and recognition of gesture. *IEEE ICCV*, 1995.

[15] C. Bregler. Learning and recognizing human dynamics in video sequences. *CVPR*, 1997.

[16] H. Buxton. Generative Models for Learning and Understanding Dynamic Scene Activity. *Workshop on GMBV*, 2002.

[17] H. Buxton. Learning and understanding dynamic scene activity: A review. *Image and Vision Computing*, 2003.

[18] L. W. Campbell and A. Bobick. Recognition of human body motion using phase space constraints. *ICCV*, 1995.

[19] L. Cao. Practical method for determining the minimum embedding dimension of a scalar time series. *Physica D*, 1997.

[20] L. Cao. Practical method for determining the minimum embedding dimension of a scalar time series. *Physica D: Nonlinear Phenomena*, 1997.

[21] L. Cao, A. Mees, and K. Judd. Dynamics from multivariate time series. *Physica D: Nonlinear Phenomena*, 1998.

[22] S. Carlsson and J. Sullivan. Action recognition by shape matching to key frames. *Workshop on Models Versus Exemplars in Computer Vision*, 2001.

[23] M. Casdagli. Nonlinear prediction of chaotic time series. *Physica D: Nonlinear Phenomena*, 1989.

[24] C. Cedras and M. Shah. Motion based recognition: A survey. *Image and Vision Computing*, 1995.

[25] A. B. Chan and N. Vasconcelos. Classifying video with kernel dynamic textures. *CVPR*, 2007.

[26] 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. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2009.

[27] J. Chen, M. Kim, Y. Wang, and Q. Ji. Switching gaussian process dynamic models for simultaneous composite motion tracking and recognition. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2009.

[28] K. M. Cheung, S. Baker, and T. Kanade. Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture. *IEEE CVPR*, 2003.

[29] CMU. Graphics lab motion capture database. http://mocap.cs.cmu.edu/.

[30] R. Collins, A. Lipton, and T. Kanade. Introduction to the special section on video surveillance. *IEEE Transactions on PAMI*, 22(8):745–746, 2000.

[31] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Trans. on PAMI*, 2003.

[32] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1*, pages 886–893, Washington, DC, USA, 2005. IEEE Computer Society.

[33] T. Darrell and A. Pentland. Classifying hand gestures with a view-based distributed representation. *NIPS*, 1993.

[34] J. Davis and M. Shah. Toward 3-d gesture recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 13:381–393, 1999.

[35] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. *IEEE International Workshop on VS-PETS*, 2005.

[36] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto. Dynamic textures. *IJCV*, 2003.

[37] A. A. Efros, A. C. Berg, G. Mori, and J. Malik. Recognizing action at a distance. *IEEE ICCV*, 2003.

[38] A. Elgammal, R. Duraiswami, D. Harwood, L. S. Davis, R. Duraiswami, and D. Harwood. Background and foreground modeling using nonparametric kernel density for visual surveillance. In *Proceedings of the IEEE*, pages 1151–1163, 2002.

[39] M. Figueiredo and A. Jain. Unsupervised learning of finite mixture models. *PAMI, IEEE Transactions on*, 2002.

[40] J. W. D. Fleet and A. Hertzmann. Gaussian process dynamical models for human motion. *PAMI*, 2008.

[41] A. M. Fraser. Independent coordinates for strange attractors from mutual information. *Phys. Rev.*, 1986.

[42] W. T. Freeman and M. Roth. Orientation histogram for hand gesture recognition. *International Workshop on Automatic Face and Gesture Recognition*, 1995.

[43] FutureLight. $r \ \& \ d$ division of santa monica studios.

[44] D. M. Gavrila. The visual analysis of human movement: A survey. *CVIU*, 1999.

[45] B. Ghanem and N. Ahuja. Phase based modelling of dynamic textures. *ICCV*, 2007.

[46] R. Gilmore and M. Lefranc. *The Topology of Chaos: Alice in Stretch and Squeezeland*. Wiley-Interscience, 2002.

[47] W. Grimson, C. Stauffer, R. Romano, and L. Lee. Using adaptive tracking to classify and monitor activities in asite. *CVPR*, 1998.

[48] A. Gritai, A. Basharat, and M. Shah. Geometric constraints on 2d action models for tracking human body. *ICPR*, 08.

[49] M. Harville. A Framework for High-Level Feedback to Adaptive, Per-Pixel, Mixture-of-Gaussian Background Models. *ECCV*, 2002.

[50] M. Herman. Understanding body postures of human stick figures. *PhD Thesis, University of Maryland*, 1979.

[51] F. L. Hitchcock. The distribution of a product from several sources to numerous localities. *Journal of Math. Phys.*, 1941.

[52] D. C. Hogg. Interpreting images of a known moving object. *PhD Thesis, University of Sussex*, 1984.

[53] W. Hu, X. Xiao, Z. Fu, D. Xie, and S. Maybank. A system for learning statistical motion patterns. *TPAMI*, 2006.

[54] P. Huang, A. Hilton, and J. Starck. Human motion synthesis from 3d video. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2009.

[55] O. Javed and M. Shah. Tracking and object classification for automated surveillance. *ECCV*, 2002.

[56] H. Jiang, M. S. Drew, and Z. N. Li. Successive convex matching for action detection. *IEEE CVPR*, 2006.

[57] N. Johnson and D. Hogg. Learning the distribution of object trajectories for event recognition. *BMVC*, 1995.

[58] S. X. Ju, M. J. Black, and Y. Yacoob. Cardboard people: A parameterized model of articulated image motion. *International Conference on Automatic Face and Gesture Recognition*, 1996.

[59] I. Junejo, O. Javed, and M. Shah. Multi feature path modeling for video surveillance. *ICPR*, 2004.

[60] H. Kantz and T. Schreiber. Nonlinear time series analysis. *Cambridge U. Press*, 2004.

[61] Y. Ke, R. Sukthankar, and M. Hebert. Efficient visual event detection using volumetric features. *ICCV*, 2005.

[62] Y. Ke, R. Sukthankar, and M. Hebert. Event detection in crowded videos. *ICCV*, 2007.

[63] J. F. Kenney and E. S. Keeping. *Mathematics of Statistics*. Van Nostrand, 1951.

[64] J. Kim and K. Grauman. Observe locally, infer globally: a space-time mrf for detecting abnormal activities with incremental updates. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2009.

[65] L. Kratz and K. Nishino. Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2009.

[66] W. H. L Wang and T. Tan. Recent development in human motion analysis. *Pattern Recognition*, 2003.

[67] I. Laptev. Space time interest points. *IJCV*, 2005.

[68] H. J. Lee and Z. Chen. Determination of 3d human body postures from a single view. *Computer Vision, Graphics, and Image Processing*, 1985.

[69] R. Lin, C. Liu, M. Yang, N. Ahuja, and S. Levinson. Learning nonlinear manifolds from time series. *ECCV*, 06.

[70] J. Little and J. E. Boyd. Recognizing people by their gait: The shape of motion. *Journal of Computer Vision Research*, 1998.

[71] C.-B. Liu, R.-S. Lin, N. Ahuja, and M.-H. Yang. Dynamic textures synthesis as nonlinear manifold learning and traversing. *BMVC*, 2006.

[72] R. B. M. B. Kennel and H. D. I. Abarbanel. Determining embedding dimension for phase space reconstruction using a geometrical construction. *Physics Review A*, 1992.

[73] D. Makris and T. Ellis. Learning semantic scene models from observing activity in visual surveillance. *Systems, Man and Cybernetics, Part B, IEEE Transactions on*, 2005.

[74] L. Marcenaro, F. Oberti, G. F., and C. Regazzoni. Distributed architectures and logical-task decomposition in multimedia surveillance systems. *Proceedings of the IEEE*, 2001.

[75] R. Mehran, A. Oyama, and M. Shah. Abnormal crowd behavior detection using social force model. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:935–942, 2009.

[76] T. B. Moeslund and E. Granum. A survey of computer vision based human motion capture. *CVIU*, 2001.

[77] G. Mori and J. Malik. Estimating human body configurations using shape context matching. *ECCV*, 2002.

[78] G. Mori, X. Ren, A. Efros, and J. Malik. Recovering human body configurations: Combining segmentation and recognition. *IEEE CVPR*, 2004.

[79] B. Morris and M. Trivedi. Learning trajectory patterns by clustering: Experimental studies and comparative evaluation. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2009.

[80] E. A. Nadarya. On estimating regression. *Theory Pb. Appl.*, 1964.

[81] B. North, A. Blake, M. Isard, and J. Rittscher. Learning and classification of complex dynamics. *IEEE PAMI*, 2000.

[82] A. Oikonomopoulousm, I. Patras, and M. Pantic. Spatiotemporal saliency for human action recognition. *IEEE ICME*, 2005.

[83] V. Parameswaran and R. Chellappa. View invariance for human action recognition. *IJCV*, 2006.

[84] M. Perc. The dynamics of human gait. *European Journal of Physics*, 2005.

[85] R. Polana and R. Nelson. Temporal texture and activity recognition. *Motion-Based Recognition*, 1997.

[86] L. Ralaivola and F. dAlcheBuc. Dynamical modeling with kernels for nonlinear time series prediction. In *NIPS*, 2003.

[87] R. Rashid. Towards a system for the interpretation of moving light display. *IEEE PAMI*, 1980.

[88] V. P. J. M. Rehg and J. MacCormick. Impact of dynamic model learning on classification of human motion. *CVPR*, 2000.

[89] P. Remagnino and G. Jones. Classifying Surveillance Events from Attributes and Behaviour. *BMVC*, 2001.

[90] R. Roberts, C. Potthast, and F. Dellaert. Learning general optical flow subspaces for egomotion estimation and detection of motion anomalies. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2009.

[91] M. T. Rosenstein, J. J. Collins, and C. J. D. Luca. A practical method for calculating largest lyapunov exponents from small datasets. *Physica D*, 1993.

[92] J. O. Roukre and N. Badler. Model based image analysis of human motion using constrained propagation. *IEEE PAMI*, 1980.

[93] Y. Rubner, C. Tomasi, and L. Guibas. The earth mover's distance as a metric for image retrieval. *IJCV*, 2000.

[94] P. Saisan, G. Doretto, Y. Wu, and S. Soatto. Dynamic texture recognition. *CVPR*, 2001.

[95] I. Saleemi, K. Shafique, and M. Shah. Probabilistic modeling of scene dynamics for applications in visual surveillance. *Accepted for Publication in TPAMI*, 2008.

[96] A. Schdl, R. Szeliski, D. Salesin, and I. Essa. Video textures. In *SIGGRAPH*, 2000.

[97] E. Shechtman and M. Irani. Space-time behavior based correlation. *IEEE CVPR*, 2005.

[98] T. Starner and A. Pentland. Visual recognition of american sign language using hidden markov model. *Computational Imaging and Vision*, 1997.

[99] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. *CVPR*, 1999.

[100] C. Stauffer and W. Grimson. Learning patterns of activity using real-time tracking. *PAMI, IEEE Trans. on*, 2000.

[101] M. D. R. Sullivan, J. Ahmed, and M. Shah. Action mach: Maximum average correlation height filter for action classification. *IEEE CVPR*, 2008.

[102] T. Syeda-Mahmood, A. Vasilescu, and S. Sethi. Recognizing action events from multiple viewpoints. In *IEEE Workshop on Detection and Recognition of Events in Video*, 2001.

[103] M. Szummer and R. W. Picard. Temporal texture modeling. *ICIP*, 1996.

[104] F. Takens. Detecting strange attractors in turbulence. *L. N. in Math*, 1981.

[105] B. A. Turlach. Bandwidth selection in kernel density estimation: A review. In *CORE and Institut de Statistique*, 1993.

[106] X. Wang, K. Tieu, and E. Grimson. Learning semantic scene models by trajectory analysis. *ECCV*, 2006.

[107] Y. Z. Wang and S. C. Zhu. A generative method for textured motion: Analysis and synthesis. In *ECCV*, 2002.

[108] D. Weinland, R. Ronfard, and E. Boyer. Free viewpoint action recognition using motion history volumes. *CVIU*, 2006.

[109] G. Welch and G. Bishop. An introduction to the kalman filter. *SIGGRAPH*, 2001.

[110] G. P. Williams. *Chaos Theory Tamed*. Joseph Henry Press, 1997.

[111] M. L. Y.-L. Tian and A. Hampapur. Robust and Efficient Foreground Analysis for Real-Time Video Surveillance. *CVPR*, 2005.

[112] Y. Yacoob and M. J. Black. Parameterized modeling and recognition of activities. *IEEE ICCV*, 1998.

[113] J. Yamato, J. Ohya, and K. Ishii. Recognizing human action in time sequential images using hidden markov model. *IEEE CVPR*, 1992.

[114] A. Yilmaz and M. Shah. Actions sketch: A novel action representation. *IEEE CVPR*, 2005.

[115] L. Yuan, F. Wen, C. Liu, and H. Y. Shum. Synthesizing dynamic texture with closed-loop linear dynamic system. *ECCV*, 2004.

[116] T. Zhang, H. Lu, and S. Li. Learning semantic scene models by object classification and trajectory clustering. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2009.