Human and Group Activity Recognition from Video Sequences

Kyle Stephens

Doctor of Philosophy

University of York

Computer Science

December 2016

Abstract

A good solution to human activity recognition enables the creation of a wide variety of useful applications such as applications in visual surveillance, vision-based Human-Computer-Interaction (HCI) and gesture recognition.

In this thesis, a graph based approach to human activity recognition is proposed which models spatio-temporal features as contextual space-time graphs. In this method, spatio-temporal gradient cuboids were extracted at significant regions of activity, and feature graphs (gradient, space-time, local neighbours, immediate neighbours) are constructed using the similarity matrix. The Laplacian representation of the graph is utilised to reduce the computational complexity and to allow the use of traditional statistical classifiers.

A second methodology is proposed to detect and localise abnormal activities in crowded scenes. This approach has two stages: training and identification. During the training stage, specific human activities are identified and characterised by employing modelling of medium-term movement flow through streaklines. Each streakline is formed by multiple optical flow vectors that represent and track locally the movement in the scene. A dictionary of activities is recorded for a given scene during the training stage. During the testing stage, the consistency of each observed activity with those from the dictionary is verified using the Kullback-Leibler (KL) divergence. The anomaly detection of the proposed methodology is compared to state of the art, producing state of the art results for localising anomalous activities.

Finally, we propose an automatic group activity recognition approach by modelling the interdependencies of group activity features over time. We propose to model the group interdependences in both motion and location spaces. These spaces are extended to time-space and time-movement spaces and modelled using Kernel Density Estimation (KDE). The recognition performance of the proposed methodology shows an improvement in recognition performance over state of the art results on group activity datasets.

Contents

List of Figures

List of Tables

Declaration

I declare that this thesis is a presentation of original work and I am the sole author. This work has not previously been presented for an award at this, or any other, University. Some parts of this thesis have been previously published in conference proceedings; where items were published jointly the author of this thesis is responsible for the material presented here. For each published item the primary author is the first listed author.

- Kyle Stephens and A. G. Bors, Observing human activities using movement modelling. In *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Karlsruhe, 2015

- Kyle Stephens and A. G. Bors, Grouping Multi-vector Streaklines for Human Activity Identification. In *IEEE Image, Video, and Multidimensional Signal Processing Workshop (IVMSP)*, Bordeaux, 2015

- Kyle Stephens and A. G. Bors, Group Activity Recognition on Outdoor Scenes. In *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Colorado Springs, 2016

- Kyle Stephens and A. G. Bors, Modelling Moving Region Interdependency in Video Sequences Showing Human Group Activity. In *IEEE International Conference on Pattern Recognition*, Cancun, 2016

Chapter 1

Introduction

The overall philosophy of computer vision is to use the principles of pattern recognition to enable the design and development of robust, effective algorithms for machine vision. Computer vision addresses problems where the data is uncertain but often highly structured. The highly structured nature of the data makes the task possible, whilst the uncertainty adds a degree of difficulty to the task.

Given that visual data is often noisy and the approximate nature of vision techniques, machine learning was a natural tool to aid in the development of computer vision tasks. The addition of machine learning tools led to more data-driven ways of modelling parameters and henceforth learning more robust models. The introduction of machine learning to computer vision introduced the ability to develop more sophisticated and flexible methods of learning that previously were not possible when decision techniques needed to be explicitly engineered. For example, early work such as a general purpose neural network algorithm [57] using back-propagation allowed a sophisticated and robust method of recognizing hand-written digits. A second example is [108], which used principle component analysis (PCA) to produce a simple yet very efficient face recognition algorithm. Such algorithms have been very influential and still influence work in the area to this date. The early applications of machine learning methods made the great potential of learning methods clear to researchers.

1.1 Human Activity Recognition

Human activity recognition is the problem of identifying and classifying different human actions performed in a video sequence. An example of such a human action could be run-

ning or jumping. A system can be trained on particular examples of an activity (training set) and then tested on a particular example of an activity (test set). The aim of the system is to identify the correct class of activity to which a video sequence belongs, or more generally, to identify and understand what the human is doing in the video sequence.

Human activities can range from simple atomic actions such as walking or jumping, to interactions between humans, e.g. shaking hands, or a particular type of movement in sport, e.g. a serve in tennis. Since the range in complexity of an activity can vary considerably, generally only a specific complexity of activity is focused upon. Ultimately, the goal is to be able to recognize any human activity, in any possible scenario; although this is a very difficult problem that researchers continue to address.

Considering only the simple atomic human activities such as walking or jumping, many challenges still exist despite the seemingly simple nature of the activity. For example, different people may walk very differently to others (intra-class variation), different people may also perform different activities which may appear inherently similar, e.g. one person's run may be similar to another person's jog. Other challenges that arise include: difference in camera viewpoint or video from a moving camera, occlusion due to objects or other humans in the scene, illumination changes, e.g. walking along a corridor passing a window, and so on.

From a more practical viewpoint, building a human activity recognition system introduces its own challenges. The main practical challenge is due to the sheer size of the data, data of this nature is often referred to as "big data". Consider a short video sequence with a resolution of 640×480, at 25fps and 10 seconds in duration; this short video sequence has 250 frames, each containing $307,200$ pixels, giving a total pixel count of $76,800,000$. Considering that a single dataset may contain hundreds of videos, this is a substantial amount of data to store and process. Considering the amount of data for just a single short video sequence, a human activity recognition algorithm must be efficient enough such that the system can deal with a reasonably-sized dataset in a reasonable amount of time.

One other interesting problem is of localizing activity within a video sequence. Localizing an activity is the ability to identify a region in the video sequence corresponding to a certain instance of a human activity. Localizing human activities is far more useful than classifying an entire video sequence as a particular class of activity and leads to a more useful application of human activity recognition in real-world applications, e.g. vi-

sual surveillance. For example, an unknown video (or video stream in real-time) can be "searched" through for an instance of a particular activity which may or may not exist in the video.

Many different techniques have been applied to the problem of human activity recognition, most commonly the traditional bag of video words (BoW) pipeline where regions of interest are extracted from the video sequence and then clustered to form region prototypes. Each video sequence is classified using traditional statistical classifiers such as support vector machines (SVM) according to which "region prototypes" the video sequence contains and how many of each "region prototypes" the sequence contains. Other approaches to activity recognition exist such as motion primitives, dynamic models and structural methods such as graph-based methods.

Solving the problem of human activity recognition opens up a world of potential new applications. Example applications include visual surveillance, video retrieval tasks (on video archives) and observing human patterns and behaviours for a better understanding of human behaviours.

1.2 Thesis Overview

An overview of the thesis is provided as follows:

Chapter 2 provides an in-depth survey of the significant contributions to the field of human activity recognition including group activity recognition and anomalous activity recognition. This chapter also surveys the most significant contributions in the areas of local features, short and medium term tracklets and BoW methodologies.

Chapter 3 presents a methodology for modelling simple human activities as contextual graphs of space-time features using the graph Laplacian. In this work, spatio-temporal activity regions were extracted, and features were modelled as similarity graphs across space and time. In other graph based approaches to human activity recognition, limitations were placed on the features due to issues representing and comparing the complex feature graphs. To overcome the limitations of typical graph based methodologies, the Laplacian representation of the graph was used, providing a vector-based representation of the graph while maintaining its discriminative nature. A further distinction of the proposed method is that the relationship between features was modelled; in the typical approaches to human activity recognition using BoW, the contextual and relationship between features is often ignored. While the results did not match those of the state of the art; it is suggested that

this approach is better suited to more complex activities such as human interactions and contextual group activities.

In Chapter 4, a new online activity monitoring approach is adopted based on forming a dictionary of activities and assessing a new activity using detection theory. In this approach, the original streaklines approach was extended to a block-based methodology; where streakline flows were segmented using the EM algorithm under the Gaussian modelling assumption. Segmented regions were then represented in the movement and location space by their block-based streakflow and location models. PCA was then utilised to project the principal streakline vector representing each moving region. The streakline representation was extended to a multi-vector approach where each block is represented by its magnitude and direction vectors in the polar coordinates space. Furthermore, a weighting factor was introduced to balance the contribution of the magnitude and direction vectors. A novel localisation methodology was introduced to account for the perspective distortion in the scenes by only comparing activities with each other inside a dynamic window. A further distinction of this methodology is that the dictionary of activities was generated online, thus allowing for the methodology to be used in an online system; without requiring offline training like some approaches. The proposed methodology also achieved state of the art results for localising abnormal activities in crowded scenes.

In Chapter 5, a novel automatic method for group activity recognition is proposed by modelling the inter-dependant relationship between features over time. In this work, a model was proposed to describe the discriminative characteristics of group activity by modelling the relationships between moving activity regions. The interdependent relationships of movements and locations were modelled using the symmetric KL divergence between the moving regions at particular time instances. This differs from other works in the area which only model the differences between longer term tracklets, and not the differences in movement and space over the short to medium term. A new stationary pedestrian detector was proposed to keep track of the stationary pedestrians by marking the locations when the pedestrians stop moving. In addition to modelling the differences in movement and location over time, the changes in such movement and location differences where also modelled using Kernel Density Estimation (KDE). The use of KDE showed a clear improvement over using conventional histograms. This differs from other methods which usually only consider the differences in features at a particular time, and do not model the changes in such differences over time. Experimental results on state of the art

group activity datasets show a clear improvement over state of the art methodologies.

Finally, Chapter 6 provides a extended summary of the contributions of the thesis, highlighting both the strengths and the weakness of the work. This chapter concludes with suggestions for future research work.

Chapter 2

Literature Review

This chapter provides a comprehensive literature review of human and group activity recognition from video sequences. Firstly, human and group activity recognition will be discussed on a high level, including the general problem, main challenges and its potential applications. Following this, an in-depth review of the current literature will be provided including discussions of state of the art methods and their strengths and weaknesses. Towards the end of this chapter, more specific human activity recognition tasks will be discussed including human interaction recognition, group-based human activity recognition and anomalous activity identification.

2.1 Human Activity Recognition

The overall aim of human activity recognition is to analyse and understand human motion in a video sequence. Practicably, the goal is to categorize a video sequence or part of a video sequence as a particular class of human activity. In this context, the class of human activities can vary considerably from the simple to the much more complex. From simple to complex, these include: gestures or "actoms", simple actions/activities, human to human interactions, human to object interactions and group activities. This literature review is largely focused on recognizing human activities, human interactions and group activities. A single instance of a human activity typically lasts a few seconds in duration, although with periodic activities such as walking there may be no obvious end to the activity, and a video sequence may consist of the same simple activity repeated several times. An example of such a walking activity is shown in Figure 2.1. Further examples of human activities including a gesture activity, tennis server activity and a surveillance scenario are shown

in Figure 2.2.

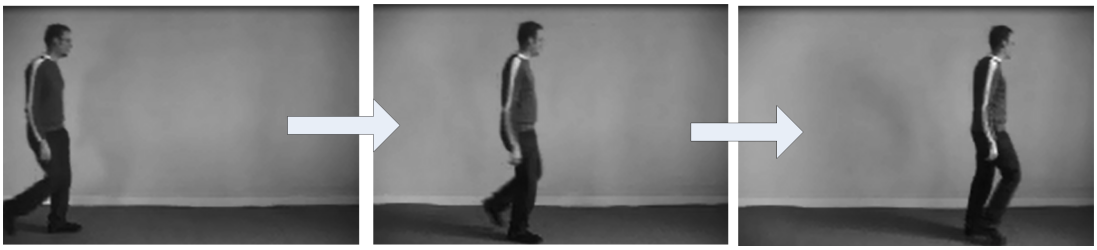
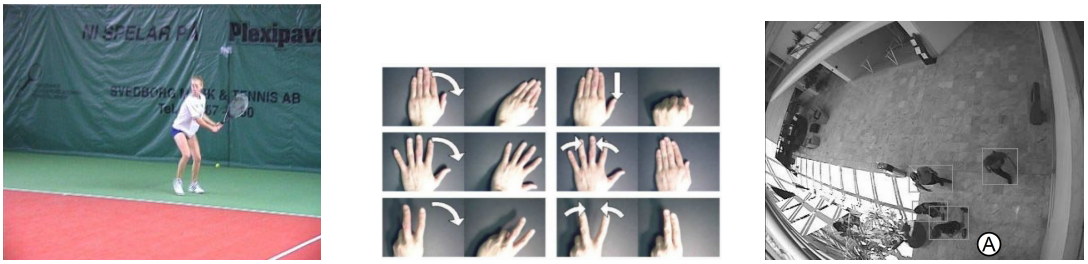


Figure 2.1: Example of the walking activity from the KTH dataset [93].



a) Tennis serve activity [123]. b) Gestures [51]. c) Activity detection [117].

Figure 2.2: Examples of human activity recognition.

2.2 Main Challenges

Due to the complexity and large variability in human actions, human activity recognition remains a very challenging task in computer vision. The main challenges in human activity recognition are:

- **Intra-class and inter-class variation of human activities** - There are many ways to perform a simple action, for example, people have may have very different gaits, wear different clothing (e.g. different textures or fittings) and walk at different paces. On the other hand, variations between classes of activities can also be problematic; different people may perform activities very different from one another. A model of human activity recognition must be general enough to model all possible examples of a particular activity yet discriminative enough to be able to distinguish between types of activities.

- **Viewpoint variation** - The viewpoint of the human is rarely the same across different scenes, for example, in one scene a person may be facing directly at the

camera, whilst in another scene, the person might be side-on to the camera. The other difficulty is the difference between aerial view and ground viewpoints; this is a common scenario in a visual surveillance system where multiple cameras monitor the same area.

- **Illumination changes** - Illumination may change a great deal between different environments, or even in the same environment due to the scene being only partially exposed to a certain lighting source. For example, scenes in an outdoor environment have very different illumination properties to scenes indoors.

- **Camera movement/jitter** - When considering a static, fixed camera this is rarely a problem; but in some real-world scenarios it may well be an issue, for example, a surveillance camera attached to a large pole may suffer from some movement in windy conditions. Another such example is video recorded from hand-held devices such as camcorders or smart-phones. Generally, stabilization algorithms can reduce the effects of camera movement, although of course no stabilization algorithm is perfect in this regard and some artefacts may still propagate through and affect the human activity model.

- **Complex dynamic backgrounds** - In real world scenarios, humans are rarely alone in a scene and against plain, easily distinguishable backgrounds. Consider a scene from a surveillance video of a shopping centre where many shoppers exist surrounded by objects and other humans all moving in different directions - in this case it is very difficult to distinguish and localize the movement of a single person.

- **Partial or full occlusions** - In a complex real-world scenario a person often walks near, behind or along objects which occludes part or all of the human body. A person may also be occluded by another human being; in this case there may be some confusion as to which person is performing the activity.

- **Noise and video compression artefacts** - When videos are compressed, noise and artefacts are naturally introduced to the videos. The video compression artefacts present in the videos are not too much of an issue in activity recognition, especially considering that many of the existing methods rely on low-resolution down-sampled videos with spatio-temporal features that rely on smoothing the regions of significant activity. Noise may be an issue in applications such as surveillance videos where the

resolution is already low and digital zoom is heavily relied upon.

2.3 Potential Applications

Despite the challenges highlighted above, successfully recognizing human activities leads to many potentially useful applications. Some examples of such applications are highlighted below:

- **Visual Surveillance -** The increasing use of video surveillance technology leads to a significant increase in the amount of video data and the need for visual analytic software. Surveillance systems are now part of modern day life in towns and cities across the world. Video analytics are helpful in such surveillance systems, especially the use of human activity recognition. Such video analytics applicable to human activity recognition could include detecting potential burglaries, thefts or scenes of violence. More generally human activity recognition could be used for car park surveillance (tracking pedestrians entering/leaving their vehicles) and monitoring of sterile/"no-go" zones.

- **Human-Computer Interaction -** HCI is now commonplace in a wide variety of modern technology systems, especially in video games and home entertainment systems. As modern technology becomes increasingly sophisticated, activity recognition becomes an increasingly useful tool. For example, gesture recognition for controlling home entertainment systems (such as televisions), or activity understanding for video game immersion.

- **Video retrieval/search -** Video archives have vastly increased in size in recent years due to video sharing websites such as YouTube. Most videos in this context are not annotated and are manually categorized by the uploader. With human activity recognition, video archives could be automatically categorized based on their content and even automatically annotated depending on the context. This leads to very useful applications in video retrieval, especially for news and sport. For example, automatically annotating a news broadcast or sports game means it is much easier to retrieve particular events or happenings at a later date.

- **Gesture recognition -** Gesture recognition can be considered a sub domain of activity recognition where the goal is to understand human gestures, that is the

movements of body parts, especially the arms and hands. Gesture recognition is heavily used for sign language recognition [48].

- **Human behavioural understanding -** Human activity recognition can be used to better understand human behaviour and to detect patterns by understanding and tracking humans in everyday scenarios. Such applications can be very useful for driving research in other areas such as sociology and urban planning. For example, in urban planning, shopping centres or town centres can be better designed if it is better understood how humans use the area. In the sociology context, it can prove useful as a study of "why humans do things the way they do", and as more specific studies of particular patterns of human behaviour.

2.4 General Human Activity Recognition Model

Recent research in the area of human activity recognition has largely focused on statistical methods using spatio-temporal features. The typical model consists of spatio-temporal interest-points which are detected in the video sequence and the local maxima becomes the center point of a spatio-temporal region. Features are then extracted from the spatio-temporal region (such as features based on optical flow or gradient values) and summarized or histogrammed to form a feature descriptor. The feature descriptors are used to form a codebook, typically followed by a "bag of visual words" model adapted from statistical natural language processing. While methods based on spatio-temporal features are the most common, other methods make use of other video features such as medium term tracking, volumetric representations and graph-based features. A general overview of the human activity recognition pipeline is shown in Figure 2.3.

2.5 Space-Time Approaches

Space-time approaches model a human activity as a 3D video volume in space-time or by a set of features extracted from the video volume. Consider an image as a matrix consisting of pixel intensity values representing the image, the 3D video volume is therefore a concatenation of the 2D images in chronological order, i.e. along the temporal dimension. A video sequence containing an execution of a human activity can therefore be represented as a 3D XYT video volume.

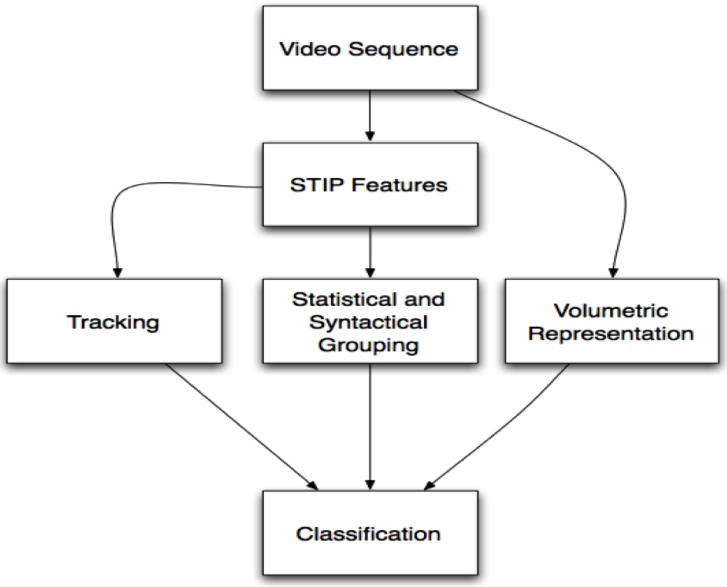


Figure 2.3: General model of human activity recognition.

Given a set of training videos, a 3D space-time volume is constructed modelling each activity from the training set. Given the 3D space-time volume of the test activity, the volume is compared with each activity model in the training data based on their similarity in shape and/or appearance. In addition to the pure 3D volume representation (essentially template matching), several variations of space-time representations exist. For example, the activity may be represent as a series of features extracted from space-time regions or as interest points extracted from the 3D space-time volume. The activity could also be represented by trajectories, where interest points detected in the video sequence are tracked over time.

Activities are recognized by matching the space-time volumes, trajectories or local features. Matching can be performed by template matching as explained previously, by neighbour-based matching where a portion of trajectories/local features are matched and some may be discarded or by statistical modelling algorithms.

Volumetric methods involve recognizing activities by measuring the similarity between two 3D space-time volumes. Instead of using the whole volume, some methods only consider the foreground regions representing the human (silhouettes) or a stack of these foreground regions. The silhouettes can be then be compared by tracking their shape changes over time.

Bobick *et al.* [8] proposed one of the earliest activity recognition methods using template matching. In this approach, each action was represented by a 2D template composed

of a 2D binary motion-energy image (MEI) and a motion history-image (MHI). The MEI is a 2D binary image which indicates where motion occurs and the MHI is an extension describing the silhouettes motion over time. Examples of both the MEI and MHI are shown in Figure 2.4. The main drawback of using MHIs is the overwrite scenario (or self-occlusion); that is, where a motion is repeated over the same spatial space as a previous motion thus overwriting the original motion. The images are constructed by 2D projections of the original 3D space-time volume. Two templates are compared using Hu moments. In their approach they were able to recognize very simple activities such was arm waving and sitting down. Their system was also applied in real-time to a children's play environment named "Kids-Room".

Figure 2.4: Example of the motion energy image (MEI) and motion history image (MHI) from [8]. Image from [8].

An extension of the previous MHI method was proposed by [3] to avoid overwrites. In this method, four optical flow channels are used (horizontal and vertical components each with positive and negative directions) to avoid self-occlusion of the person. These optical flow channels were originally proposed in [31]. The optical flow vectors were also used in [4] to derive a number of kinematic features. Features include, divergence, symmetry, etc. PCA is applied to the features to determine the dominant kinematic modes.

One disadvantage of silhouette based methods is the difficulty in extracting robust, accurate silhouettes from the space-time volume. A common approach to develop a more robust silhouette is to apply the Radon (R) transform to the silhouette [114]. The R transform provides a scale and translation invariant representation of the silhouette. R transform is also used in [101] where a third dimension is used (time). A further representation that may attenuate the typical disadvantages of silhouettes is using a contour

based method. Such an example is in [98] where the human body is represented as a star skeleton which describes the angles between reference lines of the joints, e.g. how far the hand is from its reference line. Finally, [112] proposed combining both contour based features and silhouette features for a more robust representation of human activities.

Schechtman *et al.* [97] estimated motion flows of a 3D space-time volume with application to activity recognition. In this method, they computed a 3D space-time template of the activity composed of space-time patches extracted at each location. Matching is performed by matching space-time patches in the template video to patches in the test video. Each local space-time patch represents a flow of a particular local motion in the video. Each local patch adds a score to the system. The scores are aggregated to form an overall correlation measurement between the template video and the test video. The system was successfully able to recognize activities in the Weizmann dataset and on video sequences from the 2004 Olympic Games, for example, pool dives.

Ke *et al.* [49] also used spatio-temporal volumes to model human activities. In this method, hierarchical mean-shift is applied to the volume to cluster similarly colored voxels to obtain segmented spatio-temporal volumes. The volumes are deliberately over-segmented and recognition is performed by searching over the volume for a portion of the spatio-temporal segments that match those in the activity model. Their system was successful in recognizing activities from the KTH dataset and also recognizing activities in video sequences from a TV broadcast (tennis plays).

Rodriguez *et al.* [87] proposed to recognize human activities using space-time volumes by synthesizing filters. In this method, maximum average correlation height (MACH) filters (common for image analysis) are extended to the 3D case, i.e. 3D space-time volumes. For each activity class a single synthesized filter is generated from the video volumes. Recognition is performed by applying the synthesized activity filter to the test sequence and observing its response. The MACH filters were also extended to vector values using the Clifford Fourier transform. The method was successful in recognizing activities on both the KTH and the Weizmann dataset and also on their own dataset consisting of simple activities. Examples of the 3D MACH filters are shown in Figure 2.5.

The main disadvantage of space-time volume-based representation is that much of the data captured and modelled by the method is not salient; unlike spatio-temporal features where only the most salient interesting regions are modelled. Another disadvantage is that by using the entire video volume the computational complexity of the method generally

Figure 2.5: Example of the 3D MACH filters applied to activity recognition from [87]. Image from [87].

increases quite significantly.

Baktashmotlagh *et al.* [6] applied non-linear stationary subspace analysis (NLSSA) to activity recognition. HOG features were utilized to describe the video volume. Their method relies on the fact that standard dimensionality-reduction techniques fail to account for the fact that only part of the signal is shared across all classes. NLSSA overcomes this issue by separating the stationary and non-stationary signal that is shared across all videos. That is, modelling the parts that are shared across all videos. This method removes the instant-specific information from the videos which usually introduces noise to the classification. This approach was applied not only to activity recognition but also to dynamic texture classification and scene recognition. The approach achieved better than state of the art results on both the KTH and the UCF Sports dataset.

The methods discussed in this section use local features extracted from 3D space-time volumes of a video sequence to represent an activity. The aim of these methods is to extract local features that describe the characteristics of the activity. The features are then matched across video sequences to recognize activities.

The approaches in this section have three important aspects: how and what features are extracted, how the features are represented, and finally how the features are classified. The approaches firstly detect and extract a number of local features capturing the motion of the activity. Secondly, the local features are described using a feature descriptor and the local features are combined either by the BoW paradigm (ignoring relations) or by considering their spatio-temporal relationships in some way. Finally classification is performed, usually using conventional statistical classification techniques such as SVM.

Zelnik-Manor *et al.* [123] proposed to learn dynamic events from video sequences using local space-time features extracted at multiple scales. Histograms of gradient-based space-time features are used to represent a video sequence. A simple statistical histogram-based

measurement is used to measure the difference between behaviour in the video sequence. Finally a clustering algorithm is applied to the histograms to recognize human activities. Simple human activities performed outdoors were recognized such as playing basketball or tennis.

Blank *et al.* [40] calculated local features for each frame in the video sequence. In this method, they calculated appearance based features which are obtained via solutions to the Poisson equation. The Poisson equation has shown to be very useful in extracting local shape information, particularly for object recognition. Each video sequence is represented as a set of features which are the weighted moments of the local features. This method was successfully applied on the Weizmann data with good recognition results.

Several approaches have utilized the use of sparse-interest points (or sparse local features) to recognize human activity. Well known previous local feature detectors for 2D images include scale-invariant feature transform (SIFT) and the Harris corner detector. The Harris detector was extended to the 3D case (space-time video volume) by [55]. In [55] they recognize human activities by extracting space-time interest points from video sequences. The interest-point detector detects corners in the 3D space-time volume usually capturing types of non-constant motion patterns. They also proposed a new dataset named the KTH dataset consisting of simple action videos. The new dataset was widely adopted and is discussed in more detail in [55]. Following this work, many researchers adopted the paradigm of extracting space-time interest points for activity recognition and many more feature detectors were developed.

Dollar *et al.* [29] proposed a spatio-temporal feature detector based on extracting cuboids at regions of significant activity. Their detector is based on detecting regions of significant activity from a 2D Gaussian smoothing kernel along the spatial dimension and a quadrature pair of 1D Gabor filters along the time dimension. The detector responds strongest to regions which contain periodic motions such as waving. They evaluated several different descriptors for the cuboids and found that a simple concatenation of the brightness gradient values (followed by PCA for dimensionality reduction) achieved the best performance. A codebook of cuboid prototypes is constructed by clustering the cuboids using the k-means algorithm. Finally each activity is modelled by a histogram of cuboid types detected in 3D space-time, ignoring any relationships between cuboids (BoW paradigm). The method was used not only to recognize human activities (KTH dataset) but also for facial expression recognition and mouse behaviours. Both the Dollar

and Laptev detectors have been widely adopted in many approaches to human activity recognition.

Rapantzikos *et al.* [85] extended the cuboid features to include color and motion information. Liu *et al.* [62] proposed to prune cuboid features to choose the most significant, robust features (both motion and static features). They utilized PageRank to mine the most informative static features. Bregonzio *et al.* [14] also proposed a cuboid selection method similar to [62].

More recently, Kumar *et al.* [54] used a simple optical flow based approach to human activity recognition by using the optical flow vectors along the edges of the action performer. These vectors formed feature descriptors which were passed to a multi-class SVM classifier. In their work, state of the art results were achieved (on the Weizmann and KTH datasets) while maintaining a simple and efficient approach.

Niebels *et al.* [77] proposed a new method for activity recognition using the feature detector proposed by Dollar [29]. Their method is a generative method using probabilistic Latent Semantic Analysis (pLSA). pLSA is commonly used in text mining but in this case has been used to model and recognize human activities. Features in the scene are placed into categories depending on their posterior probability of being generated by an activity. Their method was used to recognize simple actions from the KTH and Weizmann dataset.

Since the introduction of the cuboid detector and Laptev's detector, many other feature detectors and descriptors have been proposed. For example, Samanta *et al.* [91] proposed using a 3D facet model to detect STIPs named "FaSTIP". Williems *et al.* [118] proposed using the determinant of a Hessian matrix as the saliency measure for feature detection and Scovanner *et al.* [94] designed a 3D version of the classic SIFT descriptor. As explained earlier, generally these methods are used as the first step in a BoW-style pipeline. However, in using the BoW paradigm the spatial and temporal relationships between interest points are ignored. The methods discussed so far do not utilize any spatio-temporal relationship information among the features. Although such methods may prove very successful in recognizing simple periodic activities, they struggle to recognize activities in more complex scenes where the spatio-temporal relationships are much more significant.

Another issue with the BoW model is that the optimal number of "video words" must be found. Liu and Shah [95] applied maximization of mutual information (MMI) for visual word generation to automatically discover the optimal number of video word clusters. Compared to methods such as k-means (the typical clustering method for BoW-

based approaches), MMI is able to produce a higher level of word clusters, which are more meaningful and more discriminative.

Dhar *et al.* [28] aimed to overcome the shortcomings of low-level features by introducing a Directive Local Binary Pattern (DLBP) feature which incorporates orientation information with intensity differences of binary silhouette images. These features are then further combined with Edge Orientation Histograms (EOH) to form a distinctive mid-level feature representation. Experiments were performed on a range of videos containing various moving humans and the outcomes of the method were encouraging.

Similarly, Abdelhedi *et al.* [1] used a mid-level feature approach by constructing a discriminative model combining optical flow with Hu and Zernike moments. On the low level, motion vectors are extracted by forming motion curvatures. Secondly, the Hu moment and Zernik are determined which serve as a second feature vector of the activity. The resultant feature were fed into an Artificial Neural Network classifier (ANN), with good results on the Weizmann and KTH datasets.

In the approaches mentioned so far, spatio-temporal relationships between the spatio-temporal interest points have been ignored, but recently, the spatial and temporal configurations of regions has received an increasing amount of attention, especially for recognizing more complex activities. These methods attempt to model the spatio-temporal relationship between spatio-temporal interest points.

Savarasse *et al.* [92] proposed a method to include the spatio-temporal proximity information between the features. In this method, they measured feature co-occurrence patterns from a local space-time region, constructing histograms called St-correlograms. Similarly, Laptev [56] constructed space-time features by dividing a space-time volume into grids. Spatio-temporal histograms were producing measuring how the features are distributed in space-time by analysing which features fall into which grid. The method was evaluated on both KTH dataset and scenes from movies with successfully results. Finally, Lui *et al.* [61] also considered the correlations among features.

Ryoo and Aggarwal [90] introduced a method named spatio-temporal relationship match (STR match). This method explicitly models spatial and temporal relationships between features. The method aims to model the structure similarity between the video sequences by considering the spatio-temporal relationships among spatio-temporal interest points. The method was successfully able to recognize activities from the KTH dataset and also able to recognize more complex activities such as human-to-human interaction

activities. Examples of the STR match approach are shown in Figure 2.6.

Xu *et al.* [120] proposed a new hierarchical spatio-temporal model (HSTM) which used a two-layer hierarchical classification model. The bottom layer aims to capture the spatial relations in each frame, which the top layer utilises these learned features to characterise temporal relationships across the video sequence. The main advantage of such a method is that both the similarity in spatial and temporal context are well captured.

Wang *et al.* [113] also proposed a hierarchical approach, based on the existing human memory model. In this case, a context-associative approach was used to recognise human-object interactions. The system parsed high-level activities into consecutive sub-activities, followed by building a context cluster to model the temporal relationships. A series of similarity functions were used to define the retrievals over a contextual memory, similar to the auto-associative characteristics of human memory.

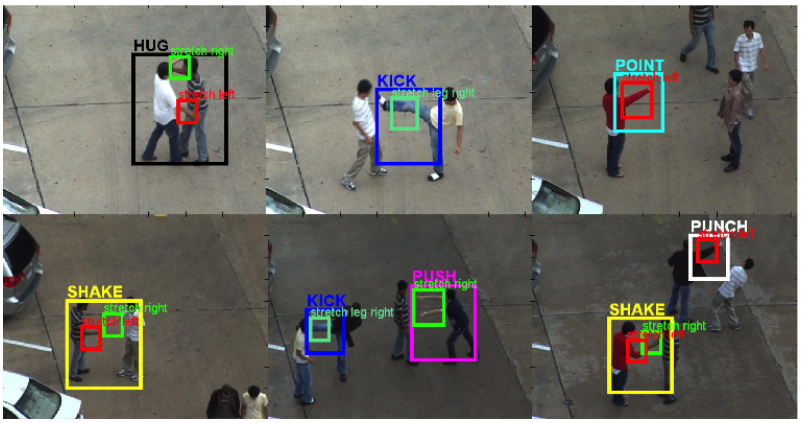


Figure 2.6: Example of the STR match approach from [90]. Image from [90].

Niebles *et al.* [76] proposed a framework for modelling motion by considering the temporal structure of the activities. Activities are represented as temporal compositions of motion segments. The model encodes the temporal compositions together with appearance motion models for each segment. Recognition is performed based on the quality of matching the model according to the appearance motion model and motion segment compositions. The method successfully recognized activities from the KTH dataset achieving state-of the art recognition results. They also introduced a new dataset consisting of complex Olympic sport activities and evaluated their method on the new dataset with good results.

Gaidon *et al.* [36] proposed to model activities as a sequence of atomic action units called “actoms”. Actoms are semantically meaningful parts of an activity that are char-

acteristic of the action in some way. The actom sequence model (ASM) represents the activity as a sequence of histograms of actom features. ASM can be considered a temporal extension of the bag-of-features paradigm. One disadvantage of this method is that it requires the manual annotation of actoms in the training set. The method was evaluated on the challenging "Hollywood-2" dataset achieving state of the art results.

The space-time approaches that use local features have several advantages. Firstly, background subtraction is not usually required. Secondly, the features are usually scale, rotation and translation invariant. The main disadvantage of space-time local features is the difficulty in modelling the structure between the local features. Due to the success of the bag-of-features approaches in recognizing simple activities, this is not strictly an issue for simple periodic activities. As datasets become more and more challenging modelling the structure and spatio-temporal relationships between the space-time local features becomes more and more important.

Trajectory-based approaches model the activity as a set of space-time trajectories. The trajectories can be considered as a set of points in 2D or 3D space tracked over time. The points could correspond to the positions of human joints, for example, when used in conjunction with body part estimation to extract the joint positions of a person at each frame. The points could also be the position of features obtained by tracking space-time features over time.

Early work by [16] recognized human activities by representing them as trajectories in phase spaces. A 3D body part model was used to track the joints of the person. The 3D XYZ body-part model at each frame was used to construct the trajectories in phase spaces. The body phase spaces are a space where the axis relates to an independent part of the body, e.g. knee-angle. An action corresponds to a set of points in the phase spaces. Finally, the trajectories from the phase space are projected into 2D subspaces and the projected trajectories are used to represent the activity. The most robust trajectories from the 2D subspaces are used for activity recognition. This method was applied to recognize basic ballet movements with marks attached to the person to track the joint positions in time.

Rao and Shah [84] proposed to model human activities by extracting curvature patterns from trajectories. They tracked the positions of the human hand by using skin pixel detection on 2D images. The tracked position over time forms the trajectory curves representing an activity. Learning was possible by constructing several action prototypes

(trajectory curves) representing the human activities. The action prototypes are essentially templates and can be matched to trajectory curves extracted from an unknown (test) video. Their approach was successful in recognizing human activities in an office environment. They also showed that their trajectories are view invariant, proof of which is shown in the paper.

Several other methods use the tracking of 3D body parts for activity recognition, for example [34] and [35]. In [98], activities are represented by joint trajectories in 4D space (XYZT). Sheikh and Shah [98] used 4D XYZT trajectories to model human activities, except this method was based on using moving cameras. The main issue with such joint tracking methods is that robust joint tracking is still largely an unsolved problem in computer vision. Early work such as [47] suggested that tracking of the joints alone (i.e. human skeleton) is sufficient to model human activities.

Messing *et al.* [67] proposed to recognize human activities using the velocity histories of tracked key-points. They use a generative mixture model (GMM) to model the human activities. The velocity history feature is extended by combining the velocity history information with other local feature information such as appearance, position and high level semantic information. They also introduced a new high resolution challenging human activity dataset focusing on activities of daily living. The method was evaluated on their new dataset and outperformed other state of the art methods on their dataset. The method also performs comparably to state of the art methods on the KTH dataset.

Aside from joint tracking, trajectories have also been used for modelling human activities based on feature points. Sun [102] used SIFT-based trajectories to model human activities. In this case, they also used contextual information in ascending levels of abstraction: a point based descriptor, trajectory transition descriptor and a trajectory proximity descriptor.

Wang *et al.* [111] proposed a trajectory based method called dense trajectories which works by performing tracking on dense patches extracted at multiple scales. Feature points are sampled from a dense grid and tracked over time using a dense optical flow algorithm. They also introduced the motion boundary histogram (MBH) feature descriptor based on the derivatives of optical flow. An example of the dense trajectory sampling and MBH descriptor is shown in Figure 2.7. The method was extensively evaluated on complex datasets and outperformed state of the art methods. Jiang *et al.* [46] extended the dense trajectories approach to use local and global reference points to model the motion of dense

trajectories. One major drawback of dense sampling is the computational cost of computing the vast amount of trajectories. Another extension to dense trajectories was proposed in [80] which aims to reduce the computational cost of the point tracking. In [80], a motion boundary based dense sampling strategy is used which greatly reduces the number of trajectories while preserving the discriminative power. They also introduce novel descriptors which describe the spatio-temporal context of the motion trajectories. The method was evaluation on the KTH, YouTube and HMDB51 databased and the method significantly reduces the computation cost of the original dense trajectory approach without a reduction in performance. The method also outperforms state of the art methods on these datasets while using their spatio-temporal context descriptors.

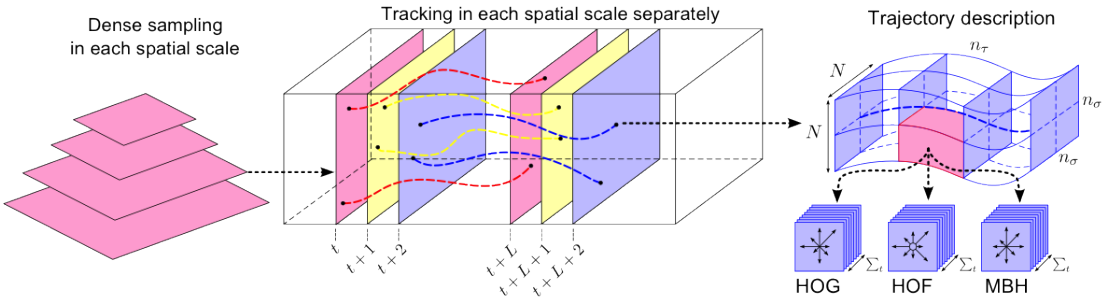


Figure 2.7: Example describing the method of dense sampling from [111]. Image example from [111]

The major advantage of space-time trajectories is that movements can be analysed in a more descriptive way. However, in the case of 3D joint trajectories, a reliable low-level joint estimation method is required. In the case of dense trajectories, the main disadvantage is the computational complexity of extracting and tracking the dense feature points. This is not a problem on smaller datasets comprising of simple activities, but is a problem on large complex datasets; it also adds the limitation that real-time systems (and practical real-world applications) are very difficult to develop whilst the computational complexity of the method is so high.

2.6 Sequential Approaches

Sequential approaches recognize human activities by considering the activity as a sequence of features. Given a set of sequential features, recognition is performed by analysing the video for a certain sequence or part of a sequence corresponding to that activity. Sequential approaches are divided into two main categories: example based approaches and state

based approaches. Example based approaches describe classes of human activity by using a training sample as a sequence of templates. That is; recognition is performed if the sequence of action executions (features) can be found in a video sequence. On the other hand, state-based approaches model the activity as a sequence of states with associated probabilities. States in this context usually correspond to a particular feature or small motion of an activity (gesture). Recognition is performed by calculating the probability that a sequence is generated in a video.

Example-based approaches represent human activities as a template of action executions. Given an unknown video, the feature vectors extracted from the video are compared to the template (example of action executions) and if the system observes a high similarity between the template and the feature vectors then the unknown video is classified an execution of that template. Since humans rarely perform activities at the same rate, any sequential methods must account for the variation in execution rate of activities.

The dynamic time warping (DTW) algorithm, widely used in speech in speech processing has been adopted for matching sequences of feature vectors for activity recognition. Early work by Darrel and Pentland [27] proposed to use DTW for gesture recognition. They modelled gestures as template images obtained from varying conditions. The correlation scores between the image frames and the template images are modelled as a function of time. The scores of the training videos are used to form the gesture template. The DTW algorithm is used to match a new observation with the templates. The DTW algorithm accounts for variation in the execution rate of the activities.

Gavrilla and Davis [38] also used the DTW algorithm to recognize human activities by using 3D body part tracking and modelling. The aim of the method was to model the skeleton of the human at each frame and model the variation in movement over time by tracking. This method was also used for gesture recognition and gestures such as waving were recognized. Yacoob and Black [121] treated the video as a set of signals describing changes of feature values. Singular value decomposition (SVD) was used to decompose the signals into a a set of eigenvectors which forms the activity basis. A test video is recognized by calculating the similarity between the input and the activity basis by calculating the coefficients of the activity basis. The method was successful in recognizing basic activities such as walking.

Other example-based methods include [31], where activities were recognized from a distance (where a human is approximately 30 pixels tall) by motion descriptors calculated

at each frame. Recognition is performed by modelling the temporal difference per frame similar to [121]. Similarly, [109] modelled activities in a similar way to [121] except they explicitly modelled the inter and intra-personal executing rates. Lublinernman *et al.* [63] proposed a method to recognize human activities by modelling them as a linear-time invariant (LTI) system. The activities are represented by an LTI system which models the changes in silhouette features over time.

The state-based approaches model the activity as a sequence of activity states. The probability of the model generating a sequence of feature vectors is calculated using a similarity measurement between the model and the video input sequence. Generally, the probability is either modelled by the maximum likelihood estimation (MLE) or the maximum posteriori probability (MAP) classifier. The most widely used state-modelling techniques are Hidden Markov Models (HMMs) and dynamic Bayesian networks (DBNs).

Yamato *et al.* [122] proposed to model human activities using HMMs. In their work they represented the features by converting binary foreground images into meshes. The feature vectors are considered as an sequence of observations generated by the model. Each activity is represented by a single HMM that corresponds to a particular sequence of feature vectors. The parameters of the HMMs are trained and they are used to recognize activities by measuring the similarity between the input (test) video and the HMMs. Various activities such as tennis plays where successfully recognized using the system. Other methods using HMMs were produced such as [9] where they recognized gestures as 2D trajectories from movements of the hand. Other such examples include [78] and [73]. Oliver *et al.* [13] introduced coupled HMMs (CHMM) to model human-to-human interactions. CHMMs overcome the main disadvantage of the basic HMM which is that only one state can be active at a single time therefore it is difficult to model complex (human-to-human) activities. Natarajan and Nevatio [73] used coupled hidden semi-Markov models (CHSMMs) which extended CHMMs to also model the duration of an activity at each state.

Gao and Sun [37] modelled activities using a discriminative latent variable model using human trajectories obtain from specific motion regimes. The trajectories are modelled using Hidden Conditional Random Fields (HCRFs). Their experiments show the superiority of the model over traditional state models including HMMs.

Park and Aggarwal [79] used a DBN to recognize gestures between humans. DBNS are an extension of HMMs in which multiple hidden nodes generate observations at each time

frame. They model the gestures as a transition of nodes (poses) from each time frame to the next. Each pose has a set of features corresponding to features obtained from body parts. The features are obtained from describing the body parts, e.g. location of skin regions and orientation of body parts.

In general, sequential approaches have the ability to model a complex sequence of smaller actions which may be used to model more complex activities or gestures. The state based methods are able to calculate the probability of an action occurring which may be easily incorporated into other decision making systems. The disadvantage of state-based methods is that it is difficult to generalize the algorithm well. For example, if one state (part of an activity) is completely missing from a scene, (e.g. because of partial occlusion) then it is difficult to recognize the activity. The other disadvantage of sequential approaches is that a large number of training examples is required to be able to model the variation of the activities.

2.7 Syntactical Methods

Syntactical methods model the activities as symbols, or more specifically as a string of symbols where each symbol corresponds to an simple atomic activity or gesture. Context-fee grammars (CFGs) and stochastic context-free grammars (SCFGs) have been widely used to recognize activities. The rules imposed by CFGs lead to a natural high-level description of the activity. The atomic activities are represented by features as described in the earlier sections. Methods described in this section are often built upon the lower level methods described earlier; to provide a more higher-level representation of the activity.

Ivanov and Bobick [44] proposed to use SCFGs for human activity recognition. They modelled human activities as a set of simple atomic actions described using SCFGs. Moore and Essa [69] extended this method to focus on multi-task activities.

The advantage of syntactical methods is that they are able to model well complex activities which are formed from simple atomic activities. The main limitations of the syntactical methods is that they are limited by the atomic actions they are composed of. For example, if the video sequence does not contain the atomic actions in that particular sequence then it is difficult to recognize such an activity. The other limitations is that it is difficult to produce a set of production rules to cover all possible events. For example, an unknown video may contain an activity for which there is no production rule. Finally, the syntactical methods have the advantage of being able to be combined with a simpler

approach (e.g. space time) to model complex activities or events.

2.8 Graph-based Methods

A graph is a data structure consisting of a set of ordered pairs (edges) of certain entities called nodes. An edge is a connection from one point to another on the graph (between nodes). The graph edges may also have an associated edge value, such as a symbol or numerical attribute, for example, a cost or length. Graph-based techniques have been proposed as a powerful tool for Computer Vision, especially in applications such as image segmentation [64] and object matching [82].

In older works, graphs have rarely been used for the representation of human activities due to the difficulty of modelling the human activities. Despite these difficulties, graphs have been recently proposed to model human activities [15, 17, 32, 106].

Brendel and Todorovic [15] proposed to model human activity using spatio-temporal graphs. In their work, they define an activity in terms of temporal configurations of primitive actions. The spatio-temporal graphs model the spatio-temporal relationships between the activity parts; or more specifically, the nodes correspond to video segments (at multiple scales), and the edges capture their spatio-temporal relationships. The method was evaluated on both the Olympic and human interaction datasets with state of the art results.

Ta *et al.* [103] modelled human activities using graphs composed of sets of spatio-temporal interest points obtained using the Dollar detector [29]. Hyper graphs with 3 edges are constructed to represent the activity. An example of localising activities using these graphs are shown in Figure 2.8. Rather than trying to recognize activities by classifying the entire sequence, this method searches the video (scene graph) for instances of the model graph. This method has the advantage of not only being able to detect and localize human activities but to detect multiple instances of activities occurring simultaneously. The method was evaluated against the KTH and Weizmann datasets obtaining state of the art results.

The main drawback of graph-based representation is that as the number of nodes and edges grow, the complexity of the graph increase significantly. The second major drawback is that basic operations such as sums cannot be performed directly on graphs making them unsuitable for conventional pattern recognition classifiers.

An alternative to directly comparing graphs is to use graph embedding. Graph embed-

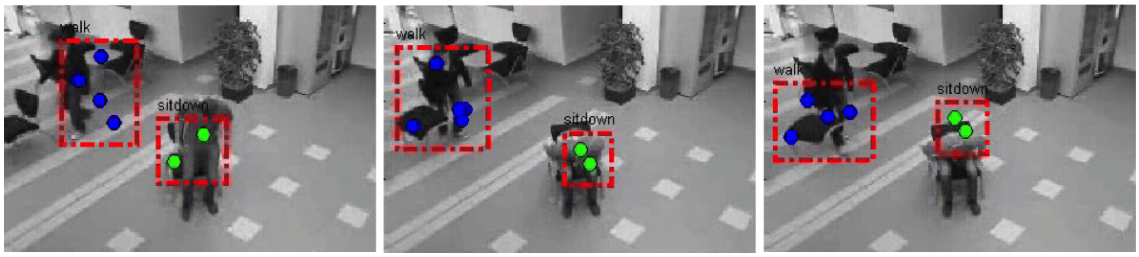


Figure 2.8: Example of localising activities using the method proposed in [103]. Image from [103]

ding converts the graph into a vector-based representation. Graph embedding offers an alternative representation which solves many of the problems listed above (such as graph matching), and allows basic operations to be performed on graphs. Once the graph is embedded into a vector-based representation it is then suitable for conventional pattern recognition approaches based on feature vectors and can be used by conventional statistical classifiers.

Graph-embedding has been proposed for activity recognition [10, 11, 32, 128]. In [32], a graph is built using the locations of SIFT key-points. The graph models the humans shape during the performing of the activity. They proposed a discriminative approach where the graph is embedded as a feature vector based on a prototype set and the probabilistic graph edit distance (P-GED). The method was evaluated on the KTH dataset and the results do not match those of the state of the art methods.

Graph-embedding has also been proposed for human activities using the silhouettes of the human body [107, 128]. In [128] a co-occurrence matrices descriptor is introduced and the shape manifold is learned using diffusion maps. Tseng *et al.* [107] also used silhouettes, but the shape model is learned using a Adaptive Locality Preserving Projection (ALPP) method and Large Margin Nearest Neighbour (LMNN) is utilized as the metric learning method. The major disadvantage of silhouette-based methods is the difficulty in extracting a robust silhouette automatically. Both methods achieve state of the art results on complex human activity datasets.

A related theme to graph embedding is spectral clustering. Spectral clustering is the study of the Laplacian representation of the similarity matrix of the data before clustering in fewer dimensions (dimensionality reduction). Spectral clustering has been proposed for the action recognition of insects [71]. In this method, the object is tracked in 3D and features are constructed of the objects 3D movement followed by the application of a

spectral clustering algorithm.

The main advantage of graphs is that they are able to capture complex visual patterns and represent them as a smart structure of features suitably connected to each other. Graphs are therefore a very powerful and flexible way to model the relationship between spatio-temporal regions of activity, or relations between parts of the human body. As activity recognition datasets continue to increase in complexity the relationship between spatio-temporal regions of activity becomes far more important.

2.9 Human Interaction Recognition

In this section, human interaction recognition is discussed. Human interaction recognition is the recognition of interactions between humans (human-to-human interaction) or between humans and objects. In the case of human-to-object interaction, the identification of objects and motion is required along with simple activity recognition for robust detection of the interaction.

Moore *et al.* [70] developed a system where object recognition is performed followed by human activity recognition. That is, the object is recognized first and then the interaction between the object and human is estimated. HMMs are used to characterize the actions. A Bayesian network is used together with object and human activity recognition to classify the activity. The approach was tested on various objects involving a single person and a single object, e.g. a person picking up a book or a phone. Similarly, Peursum *et al.* [81] proposed a Bayesian framework for labelling objects in an activity context. Their method calculated an interaction signature for each object which is essentially a set of activity recognition results involving the object. Similarly, [42] proposed a probabilistic model for human interaction recognition. The Bayesian network integrates information from the interaction with objects, e.g. appearance and human motion with the object to recognize an activity. Ryoo and Aggarwal [88] used object recognition and motion estimation to recognize human-object interactions such as stealing a suitcase. Finally, [89] proposed a probabilistic extension from their earlier work to compensate for the failure of low-level components.

2.10 Group-based Activity Recognition

In this section, the recognition of group activities is discussed. Group activities are those in which multiple persons perform activities as a group. For example, a group of people walking or a group of people carrying an object. To recognize group-based human activities, a higher level representation must be introduced which can model the activity as a composition of simpler activities, e.g. of multiple persons performing activities simultaneously. Most approaches in this area focus on a specific type of activity or activities in a particular scenario.

Gong and Xian [39] used a variation of dynamic Bayesian networks to recognize group activities. In their method, they were able to successfully recognize activities such as loading or unloading trucks. Similarly, Zhang *et al.* [124] recognized group activities in a meeting using DBNs, examples of such activities include giving a presentation and group discussion. Similarly, [26] also used DBNs with a hierarchical structure to recognize similar office activities. Ryoo and Aggarwal [89] developed a general representation for group activity recognition. They proposed a description-based approach which models various classes of group activities. They described the activities as a set of sub-events which correspond to individuals performing activities. Their method successfully recognized a range of group based activities including group based activities (i.e. marching) and group-based interaction (e.g. fighting).

Several algorithms have focused on the use of tracklets/short term tracks for group based activity recognition. For example, Ni *et al.* [75] recognised group activities using localized causalities based on manually initialized tracklets. Lin *et al.* [60] used a heat-map based algorithm for modelling human trajectories when recognising group activities in videos. Chang *et al.* [18] used a probabilistic approach to group human activity by forming various probabilities depending on the tracks between individuals using a multi-camera system. Choi *et al.* [23] proposed a framework for analysing collective group activities based on different levels of semantic granularity. Zhang *et al.* [125] addressed the problem of group event recognition by computing histograms of different features extracted from tracklets, representing localized movement in the video. Similarly, Cheng *et al.* [20] modelled group activity as a framework composed of multiple layers and Gaussian processes were used for representing motion trajectories.

2.11 Anomalous Activity Recognition

In this section, the online detection of anomalous human activities will be discussed. Unlike simple human activities, anomalous activity detection focused on detecting abnormal behaviours in a video sequence; given the known, expected behaviour of the scene. In this scenario, context is very important as what is considered abnormal may vary considerably depending on the scene. Another important difference is that most human activity recognition methodologies focus on the offline classification of a known set of activities, unlike in abnormal human activity recognition where the abnormal activity is known, and the algorithm must learn such abnormal activities online. A further difference is that the anomalous activities generally occur with low probability with respect to the normal activity.

The area of detecting abnormal activity from video sequences is a well researched area of computer vision, with a wide variety of proposed methods. In complex, crowded scenes, the general low-level approaches to feature representation are unreliable and the performance of such methods tend to degrade due to factors such as scene clutter, occlusions and general density of unsteady flow in the scene.

Recently, several notable methods for abnormal activity detection have been proposed. [58] proposed a detector that accounts for both appearance and dynamics using a set of mixture of dynamic texture models. [58] also introduced a dataset of densely crowded pedestrian walkways which consists of non-staged, realistic anomalies such as bicyclists and electric-vehicles. [105] proposed to model the optical flow using a spatio-temporal Laplacian Eigenmap to extract different crowd activities from videos. The motion patterns are clustered using k-means on the graph in the embedded space and a multivariate Gaussian mixture model (GMM) is used to represent the regular motion patterns. [52] modelled the motion patterns using GMMs using gradients as a 3D distribution. A dictionary of activity prototypes was learnt by identifying statistically similar cuboids using the KL-divergence between probabilistic models. Finally, GMM based Markov random fields (GMM-MRF) were used in [72] for abnormal activity detection.

2.12 Conclusion

The main methods discussed in the literature review have been space-time approaches. State of the art methods for complex human activity identification such as dense trajec-

tories and graph based modelling provide very good results on state of the art complex activity datasets. Although the results are very good on complex datasets, the recognition of human activities is still largely focused on treating human activity recognition as a classification problem, i.e. video sequences are classified as a particular activity and activities are not localized within the video sequence. The main future challenge is to develop or adapt the methods to detect, localize and recognize complex human activities in real-world scenes; particularly in crowded, unstructured scenes with a variety of background noises.

To conclude, this chapter has provided an in-depth literature review of human activity recognition. Firstly, the general problem of human activity was introduced and its main challenges were discussed. The usefulness of human activity recognition was discussed, together with a list of potential application areas. Finally, the chapter concluded with an in-depth literature review of state of the art methods for activity recognition.

Chapter 3

Human Activity Recognition using Graph Modelling

3.1 Introduction

In this chapter, we propose a graph-based methodology for human activity recognition. Human activity recognition has been an area of significant research over the past decade, mainly focused on simple, atomic human actions. The overall aim of human activity recognition is to analyze and understand human movement in a video sequence. The goal is to categorize a video sequence, or part of a video sequence as a particular class of human activity. In the real world, different levels of human activity exist from the simple atomic events to the more complex, which often require scene understanding. In this chapter, we focus on the recognition and classification of simple human activities in staged video sequences. A single instance of such a human activity typically lasts around a few seconds in duration, although some activities may appear periodic/cyclic, for example walking, where similar steps may be repeated several times.

The main body of research in the area of human activity recognition is largely focused on statistical methods using spatio-temporal features. The typical activity recognition pipeline begins by detecting spatio-temporal interest-points in the video sequence, then representing such interest points using local features, and finally summarising the local features as a feature vector or histogram. The feature vectors are then used to form a codebook, typically followed by a 'bag of visual words' model adapted from statistical natural language processing, where the features are clustered into groups (visual words) followed by supervised classification.

Many methodologies have been proposed for activity recognition using the 'bag of visual words' approach. Zelnik-Manor *et al.* [123] proposed to learn dynamic events from video sequences using local space-time features extracted at multiple scales. Blank *et al.* [40] calculated local features (via Poission equation) for each frame in the video sequence. The Harris detector was extended to the 3D case (space-time video volume) by Laptev in [55]. In [55] they also proposed a new dataset named the KTH dataset consisting of simple action videos. Dollar *et al.* [29] proposed a spatio-temporal feature detector based on extracting gradient-based cuboids at regions of significant activity. Rapantzikos *et al.* [85] extended the cuboid features to include color and motion information while Liu *et al.* [62] proposed to prune cuboid features to choose the most significant, robust features. Wang *et al.* [111] proposed a trajectory based method called dense trajectories by performing tracking on dense patches of optical flow extracted at multiple scales. They also introduced the motion boundary histogram (MBH) feature descriptor based on the derivatives of optical flow.

More recently, the spatial and temporal relationships between activity regions has received an increasing amount of attention, especially for recognizing more complex activities. Savarasse *et al.* [92] proposed a method to include the spatio-temporal proximity information between the features. With a similar motivation, Laptev [56] constructed space-time features by dividing a space-time volume into grids, assessing features and activities depending on their spatial-location.

In other methods, graph-based approaches have been proposed as a powerful tool for modelling relationships between spatio-temporal interest points [15, 17, 32, 106]. Brendel and Todorovic [15] proposed to model human activity using spatio-temporal graphs. In their work, they define an activity in terms of temporal configurations of primitive actions. The spatio-temporal graphs model the spatio-temporal relationships between the activity parts; or more specifically, the nodes correspond to video segments (at multiple scales), and the edges capture their spatio-temporal relationships. Ta *et al.* [103] modelled human activities using graphs composed of sets of spatio-temporal interest points obtained using the Dollar detector [29]. Hyper graphs with 3 edges are constructed to represent the activity. Rather than trying to recognize activities by classifying the entire sequence, this method searches the video (scene graph) for instances of the model graph. Graph-embedding has also been proposed for activity recognition [10, 11, 32, 128]. In [32], a graph is built using the locations of SIFT key-points. The graph in [32] models the

humans shape during the performing of the activity. In [32] a discriminative approach is proposed where the graph is embedded as a feature vector based on a prototype set and using the probabilistic graph edit distance (P-GED). Graph-embedding has also been proposed for human activities using the silhouettes of the human body [107,128]. In [128], a co-occurrence matrix descriptor is introduced and the shape manifold is learned using diffusion maps. A related theme to graph embedding is spectral clustering. Spectral clustering is the study of the Laplacian representation of the similarity matrix of the data before clustering in fewer dimensions. Spectral clustering has been proposed for the action recognition of insects [71], where the objects were tracked in 3D and features were constructed of the objects 3D movement followed by the application of a spectral clustering algorithm.

In this chapter, we propose a human activity recognition methodology using graph embedding. In this method, the most salient spatio-temporal interest points are selected using a detection methodology, and spatio-temporal features are extracted around the interest points. The spatio temporal relationships between the features are extracted by representing the relationships via the Laplacian representation of the similarity feature matrix. We also model the local neighbourhood features and the immediate neighbourhood of features to add contextual information. Eigen-decomposition is performed on the Laplacian representation of the embedded graph, to obtain its principal eigenvectors and eigenvalues.

The remainder of the chapter is organised as follows: Section 3.2 describes the motivation behind graphs for activity recognition, followed by an in-depth theoretical discussion of the proposed methodology. Section 3.3 describes the experimental results on two human activity datasets. Finally, Section 3.4 describes the conclusions of this research work.

3.2 Human Activity Recognition using Graph Modelling

A graph is a data structure consisting of a set of ordered pairs (edges) of certain entities called nodes. An edge is a connection from one point to another on the graph (between nodes). The graph edges may also have an associated edge value, such as a symbol or numerical attribute, for example, cost or length. Graph-based techniques have been proposed as a powerful tool for Computer Vision, especially in applications such as image segmentation [115] and object matching [82]. Graphs are able to capture complex visual patterns and represent them as a smart structure of objects suitably connected to each

other. Graphs could therefore be used as a very powerful and flexible way to model the relationship between spatio-temporal regions of activity, or model relations between parts of the human body.

One drawback of graph-based representations is that as the number of nodes and edges grows, the complexity of the graph increases significantly. A further drawback with modelling activities as graphs is the graph matching problem[1] As the complexity of the graph grows, exact graph matching becomes computationally infeasible. In the case of activity recognition, directly comparing graphs is unsuitable as the graphs are prone to significant noise due to the intra-class variations of human activities. One other major drawback of traditional graphs is that basic operations such as sums cannot be performed directly on graphs, thus making them unsuitable as a tool in conventional pattern recognition problems.

An alternative to directly comparing graphs is to use graph embedding. Graph embedding aims to compute an embedded version of the graph, usually as a vector-based representation. Graph embedding offers an alternative which solves many of the problems listed above (such as the graph matching problem), and allows basic operations to be performed on such graphs. Once the graph is embedded into a vector-based representation it is then suitable to be used for conventional pattern recognition approaches based on feature vectors and can be used by conventional statistical classifiers.

Graph embedding has been successfully used in applications such as optical character recognition (OCR) [43] and brain state decoding [86]. Various approaches to graph embedding exist such as decomposing the similarity matrix characterizing the graph by using orthogonal decompositions such as SVD [41], quantum commute times [33] and prototype selection [32].

A related theme to graph embedding is spectral clustering. Spectral clustering based methods have been widely used in Computer Vision [74]. Spectral clustering is the study of the Laplacian representation of the similarity matrix of the data before clustering in fewer dimensions. Spectral clustering is commonly used for image segmentation [64, 116], but has also been used for event detection [83] and the detection of unusual activity [110].

[1] Exact (sub)graph matching is NP complete, although approximations may exist for some applications.

Graphs for Human Activity Recognition

In previous research, graphs have seldom been used for the representation of human activities due to the difficulties mentioned previously. Despite these difficulties, graphs have recently been proposed to model human activities [15, 103]. Graphs are able to produce a model of higher representation power than traditional statistical methods and are able to model the structure of localized movements in a much more structured and natural way. Graphs also have the advantage of being able to describe the interdependence of several localized movements, which is often missing from conventional activity recognition methodologies. Graph-embedding has also been proposed for activity recognition [107, 127, 128].

Method Overview

Figure 3.1 outlines the key stages in the proposed activity recognition methodology. In the following, each step is discussed in more detail.

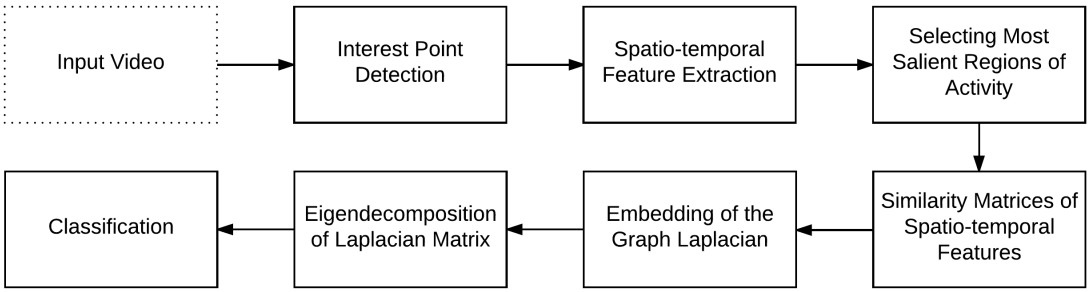


Figure 3.1: Outline of the proposed method of modelling human activities as graphs

1) **Input Video -** The input dataset form a set of videos $M_i...n$ where n is the number of video sequences.

2) **Interest Point Detection -** An interest point detector is applied spatio-temporally across the video sequences to extract the most salient interest points in the video sequence.

3) **Feature Extraction -** Features are extracted (around the interest points) from each video volume ($I_i(x, y, t)$) and a feature vector $\boldsymbol{V}_{i,j}$ is formed for each spatio-temporal activity region j in the sequence i.

4) **Selecting Regions of Significant Activity -** The m most significant regions of

activity are selected for each video sequence. The number of selected regions m is fixed across all video sequences in the dataset.

5) **Constructing Similarity Matrices -** A similarity matrix A_i is constructed for each video sequence i, based on the feature vectors $\mathbf{V}_{i,j}$ of regions of significant activity. A Laplacian matrix L_i is constructed, from A_i.

6) **Eigen-decomposition -** Eigen-decomposition is performed on each Laplacian matrix L_i. The set of eigenvectors $\{\boldsymbol{\phi}_1|\boldsymbol{\phi}_2|...|\boldsymbol{\phi}_p\}$ and eigenvalues $\{\lambda_1, \lambda_2, ..., \lambda_p\}$ are extracted from the similarity matrices, where p is the selected number of top rank (greatest magnitude eigenvalues) eigenvectors.

7) **Classification -** The eigenvectors $\{\boldsymbol{\phi}_1|\boldsymbol{\phi}_2|...|\boldsymbol{\phi}_p\}$ are concatenated for each video sequence and used for classification. Each video sequence i is categorized into a class from a pre-determined set of human activity classes, for example, running or jumping.

The proposed method has some notable advantages over traditional 'bag of words' style approaches. Firstly, the interdependent relationship between regions of significant activity are modelled, unlike traditional BoW-based approaches where the contextual and interdependent relationships are largely ignored. Secondly, the proposed method uses significantly fewer spatio-temporal features and much smaller feature vectors than traditional methods, therefore its computational complexity is greatly reduced. Finally, due to the nature of the proposed method, it could be combined with a traditional statistical method to utilise the advantages of both methods to provide a more discriminant activity model.

Feature Extraction

The first step in the proposed methodology consists of the extraction of the space-time features. Different methods have been proposed in the literature for extracting space-time interest points (STIPs) and for the description of space-time patches. Due to different detectors/descriptors been using in varying scenarios and pipelines, it is unclear if a single detector and descriptor combination perform better for human activity recognition. Some examples include the cuboid detector/descriptor [29] and Laptev's STIP detector [55] based on the Harris corner detector. Recent evaluation papers [96, 104] suggest that the cuboid detector and descriptor achieves very good recognition rates on common datasets and it is also computationally efficient. Due to this, the cuboid detector and descriptor are chosen as the detector and descriptor for this work.

The detector proposed by Dollar [29] uses separable linear filters which treats the spatial and temporal dimensions independently. A 2D Gaussian smoothing kernel $g(x, y; \sigma_s)$ is applied along the spatial dimensions and h_{even} and h_{odd} are a quadrature pair of 1D Gabor filters applied temporally. The response function is given by

$$R = (I * g * h_{even})^2 + I(g * h * h_{odd})^2 \tag{3.1}$$

and the Gabor filters are defined as:

$$h_{even}(t; , \tau_s, \sigma_s) = -\cos(2\pi t\omega)e^{-t^2/\tau_s^2} \tag{3.2}$$

$$h_{odd}(t; , \tau_s, \sigma_s) = -\sin(2\pi t\omega)e^{-t^2/\tau_s^2} \tag{3.3}$$

where I is the image, ω is a temporal parameter, σ_s and τ_s are the spatial and temporal scaling parameters, respectively. The authors in [29] suggest to use $\omega = 4/\tau$ as the number of parameters for the response function R is reduced to two. The two parameters σ_s and τ_s correspond roughly to the spatial and temporal scales of the detector and can be empirically selected depending on the resolution and nature of the video sequences. The response function R will respond strongest to periodic motions such as hand waving. The response function R will also induce a strong response at local regions where complex motion patterns are present, such as corners for example. Regions without spatially-distinguishable features and regions undergoing pure translation or motion of a constant speed will induce a low response.

Following the detection of the space-time interest points, local features will be extracted around the local maxima of R. One requirement of our approach is that a fixed number of regions is required across all video sequences in the dataset. This is due to the requirements of fixed-size graphs for comparison purposes, and also to minimize the computational complexity of the graph by only selecting the most salient regions. In order to extract a fixed number of cuboids for each video sequence, a threshold must be used on the response R to limit the number of interest points detected. We define a threshold θ, then we extract only the spatio-temporal interest points at points of R that satisfy $R > \theta$. We begin by setting θ, to a large value then slowly decrease θ until exactly n spatio-temporal interest points are detected. Following this, the spatio-temporal cuboids are extracted around the spatio-temporal interest points.

The cuboid descriptor, is a simple descriptor which is calculated by concatenating gradient values obtained along the different directions (along x, y and t). Firstly, the

spatio-temporal cube of pixels is smoothed at varying scales. Secondly, gradient values are computed along the x, y and t dimensions. Finally, the gradient values are concatenated to form a single vector of gradient values. Due to the length of the gradient vector, principal component analysis (PCA) is applied on the vector to extract the essential representation of the human activity and to reduce the dimensionality. Therefore each video sequence is now represented by its set of feature vectors, extracted from the cuboids of spatio-temporal interest points.

Similarity Matrix Representation

In this step, the proposed method aims to model the relationship between the feature vectors by modelling the interdependent relationships between feature vectors using a graph-based representation. In this work, we propose to represent the features using a similarity matrix.

Consider the set of feature vectors $\boldsymbol{X}_{1...m}$ consisting of m vectors, each feature vector $\boldsymbol{X}_i$ representing a single spatio-temporal region of significant activity. Given the feature vector $\boldsymbol{X}_i$ representing a descriptor for cuboid i, a similarity matrix A can be computed by

$$A(i,j) = e^{-\frac{\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|^2}{\sigma^2}} \tag{3.4}$$

where σ is a scaling factor which is used to weight the similarity between characteristic vectors. This results in a symmetric matrix where the values lie in the range $[0, 1]$. The matrix models the interdependent relationship between each spatio-temporal region and every other spatio-temporal region. $\|\boldsymbol{X}_i - \boldsymbol{X}_j\|^2$ models the difference between the cuboid i and the cuboid j. If $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ are statistically very different, then the result of the equation will be close to 1, comparatively if $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ are statistically similar, then the result of the equation will be closer to 0. As the scaling factor σ increases, the result tends to 1 and as the scaling factor σ decreases the result tends to 0. The values in the similarity matrices may vary considerably between different activities, therefore an appropriate value for σ must be carefully selected.

So far, only a single term (gradient) has been utilised to construct the similarity matrix. It is possible to add further terms to the equation to obtain a more discriminative representation of the activity in the video sequence. In the following we consider the relative distance between regions of activity and contextual information from the local spatio-temporal neighbourhood.

To include the distance information in the model, each spatio-temporal region extracted has a location vector $\boldsymbol{B} = \{x, y, t\}$ or $\boldsymbol{B} = \{x, y\}$ (by considering the time as a dimension), which can be used as an additional term to construct the adjacency matrix

$$A(i,j) = e^{-\frac{\left\|\boldsymbol{X}_i - \boldsymbol{X}_j\right\|^2}{\sigma_1^2} - \frac{\left\|\boldsymbol{B}_i - \boldsymbol{B}_j\right\|^2}{\sigma_2^2}} \tag{3.5}$$

where the term $\frac{\|\boldsymbol{B}_i - \boldsymbol{B}_j\|^2}{\sigma_2^2}$ models the relative distance between spatio-temporal regions. If the regions are close in space-time, the result of the term will be small, comparatively if the regions are far apart in space-time then the result will be large. As a consequence, if regions are far apart in space-time and have very different gradient values then the result of both terms will be much larger, comparatively, if regions are close in space-time and are similar, the result of the terms will be much smaller. This is a complementary effect as regions that are very different tend to be further apart in space-time and regions that are similar tend to be closer in space-time. As a consequence of adding an additional term to the equation, a second scaling factor is introduced σ_2. The two scaling factors (σ_1 and σ_2) must be carefully balanced to avoid one term becoming too dominant. σ_1 will remain the same empirically selected value, and σ_2 will be chosen dependant on the dimensions of the video space-time volume.

Modelling Spatio-temporal Contextual Information

The local neighbourhood of each region can provide some useful contextual information to provide a more discriminative representation of the human activity. For example, neighbouring cuboids may be part of the same localised activity. Similarly, neighbouring regions may provide distinguishing characteristics that aren't present in the detected cuboid. We consider modelling the local neighbourhood in two ways: using the contextual information from nearby cuboids in space-time and secondly by considering the immediate neighbouring regions (not interest points) of the cuboid.

The local neighbourhood can be considered as the closest significant regions (cuboids) in space-time. A visual example of the cuboids in space-time is shown in Figure 3.2. We define the closest significant cuboids as those that are closest in space-time by considering the Euclidean distance between cuboids. Given the location vectors of two regions of significant activity $\boldsymbol{P}_i = \{x_i, y_i, t_i\}$ and $\boldsymbol{P}_j = \{x_j, y_j, t_j\}$ the distance can be calculated by

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + \lambda_T (t_i - t_j)} \tag{3.6}$$

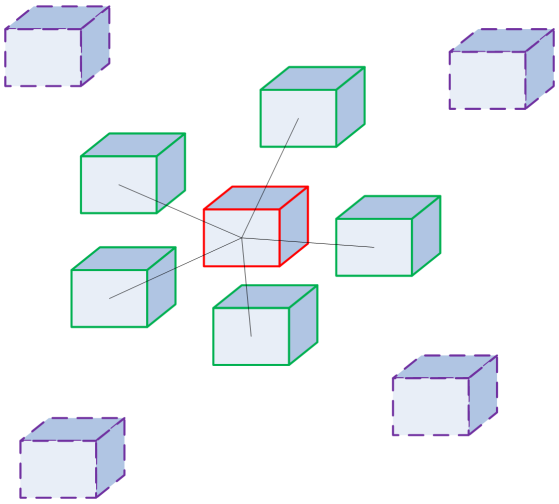


Figure 3.2: Local spatio-temporal neighbourhood of the spatio-temporal region.

where λ_T is a scaling parameter to balance the difference in scales between space and time. The value λ_T depends on the dimensions of the video volume for the given dataset as well as the frame rate. For each region, the closest n_{xy} neighbouring regions are found. Where n_{xy} is a fixed value across all the video sequences. Given these neighbouring regions, an $n_{xy} \times n_{xy}$ adjacency matrix is calculated using the same equation (3.4), which will be later combined with can be combined with the other matrices, for example, the feature matrices.

Next, we consider the immediate neighbourhood of the region. The immediate neighbourhood can be considered as regions immediately next to the original region, and of equal size to the region. We consider six immediate neighbours: four along the spatial dimensions - above, below, left and right; and two along the time dimension - before and after in time. A visualisation of the immediate neighbourhood is shown in Figure 3.3.

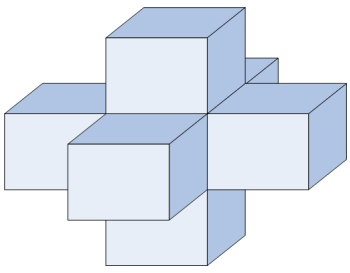


Figure 3.3: Immediate neighbourhood of the spatio-temporal region.

For the immediate region, new regions are extracted around the significant region to form new descriptor vectors $\boldsymbol{Q}_{1...7}$. Note that $\boldsymbol{Q}_1$ is the vector describing the original

significant region of activity. A similarity matrix of size 7×7 is calculated for each significant region in the same way as the local neighbourhood, using equation (3.4). By modelling the immediate region, the nearby contextual information of local regions is considered, which would previously be ignored as such nearby regions would not be considered significant. The immediate neighbourhood approach also aims to address the issue where each human movement (or each moving part) can rarely be confined to a single region thus modelling the immediate neighbours adds potential information regarding the before/after movements or other movements in the similar space which may also be of interest.

Laplacian-based representation

So far, the features have been modelled independently as a similarity matrix. The similarity matrix is one the simplest forms of graph representation, and a more discriminative representation of the graph is possible by considering the graph Laplacian. We discuss two potential representations of the graph: Combinatorial Laplacian and the normalized Laplacian matrix.

The combinatorial Laplacian is often characterised as a more useful representation of the graph than the similarity matrix as it produces a semi-definite matrix representation of the graph. Given the similarity matrix constructed from equation (3.4) or equation (3.5) and the diagonal degree matrix D, where the diagonal elements are simply the node degrees $D(u, u) = d_u$; the Laplacian matrix can be defined as the degree matrix minus the adjacency matrix

$$L = D - A, \tag{3.7}$$

where matrix L is the resulting Laplacian matrix. This matrix then represents a discrete version of the Laplacian in continuous space.

We also consider the normalized Laplacian matrix

$$\hat{L} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}, \tag{3.8}$$

Similarly to the combinatorial Laplacian representation, this produces a semi-definite representation of the graph. Due to the normalisation, the eigen-decomposition of the matrix means that the largest eigenvalue is $\leqslant 2$, and all eigenvalues are $0 \leqslant \lambda_i \leqslant 2$, where λ is the eigenvalue. Considering this, we use the normalized Laplacian matrix as the matrix representation.

Eigen-decomposition of the Activity Matrix

The spectrum of the Laplacian matrix has proved useful to characterize the properties of a graph and for extracting information from its structure. The spectrum of the graph is obtained from the matrix representation using eigen-decomposition. Eigen-decomposition will be performed on each Laplacian matrix $\hat{L}$ computed as in equation (3.8).

The spectrum is useful as a graph representation as it is invariant under similarity transform. This means that two isomorphic[2] graphs may have the same spectrum. In this case, this means that two isomorphic graphs with different orders of vertices may share the same spectrum. This is a useful property because significant regions of activity are rarely reproduced in the same order across different video sequences.

Consider the Laplacian matrix $\hat{L}$ of size $m \times m$, the eigen-decomposition is

$$\hat{L} = \Phi \Lambda \Phi^T \tag{3.9}$$

where Λ is a diagonal matrix of ordered eigenvalues (ordered by largest magnitude first) and Φ is the matrix of eigenvectors (as columns).

The eigen-decomposition is therefore the set of eigenvalues $\{\lambda_1, \lambda_2, ..., \lambda_m\}$ obtained from $\Lambda = diag(\lambda_1, \lambda_2, ..., \lambda_m)$ and the set of eigenvectors $\{\boldsymbol{\phi}_1|\boldsymbol{\phi}_2|...|\boldsymbol{\phi}_m\}$ obtained from the matrix $\Phi = (\boldsymbol{\phi}_1|\boldsymbol{\phi}_2|...|\boldsymbol{\phi}_m)$.

In our work, Singular Value Decomposition (SVD) will be used to obtain the eigenvalues and eigenvectors. SVD is a generalised method of eigen-decompositon that can be applied to any (non-square) matrix whereas eigenvalue-decomposition can only be applied to certain square matrices.

In the simplest case of a single feature matrix, represented as an $m \times m$ Laplacian matrix $\hat{L}$, the following steps are followed:

1. Eigen-decomposition (equation (3.9)) of the matrix $\hat{L}$ to obtain the set of eigenvalues $\{\lambda_1, \lambda_2, ..., \lambda_m\}$ and eigenvectors $\{\boldsymbol{\phi}_1|\boldsymbol{\phi}_2|...|\boldsymbol{\phi}_m\}$.

2. The top k_{eig} eigenvectors are selected based on the magnitude of the top k_{eig} largest eigenvalues. By selecting only the top k_{eig} eigenvectors, only the significant vectors representing the activity are kept.

[2]Graphs which contain the same number of graph vertices connected in the same way are said to be isomorphic.

3. The eigenvectors $\{\boldsymbol{\phi}_1|\boldsymbol{\phi}_2|...|\boldsymbol{\phi}_{k_{eig}}\}$ are used to represent the human activity in that particular video sequence.

In the case of a two term matrix composed of the cuboid features and the additional distance term, the steps outlined above remain the same except the first step listed above is replaced with the similarity matrix, as computed in equation 3.5.

When the second term in the similarity matrix involves either the local or contextual information, the eigen-decomposition steps change. Given the $n_{xy} \times n_{xy}$ similarity matrix N, representing the local or immediate neighbourhood information computed using equation (3.4), the following steps are followed:

1. Eigen-decomposition of the matrix N to obtain the set of eigenvalues $\{\lambda_1, \lambda_2, ..., \lambda_{n_{xy}}\}$ and eigenvectors $\boldsymbol{\phi}_1|\boldsymbol{\phi}_2|...|\boldsymbol{\phi}_{n_{xy}}\}$ representing the local/immediate neighbourhood.

2. The top k_{eig} eigenvectors are selected based on the magnitude of the top k_{eig} largest eigenvalues.

3. The eigenvectors $\{\boldsymbol{\phi}_1|\boldsymbol{\phi}_2|...|\boldsymbol{\phi}_{k_{eig}}\}$ are the selected eigenvectors used to represent the local/immediate neighbourhood.

4. The eigenvectors $\{\boldsymbol{\phi}_1|\boldsymbol{\phi}_2|...|\boldsymbol{\phi}_{k_{eig}}\}$ are used in equation (3.5) as the 2nd (or 3rd if distance is 2nd) term to construct the similarity matrix $\hat{A}$ where the first and/or second term remains the same as described above.

5. The new similarity matrix $\hat{A}$ is used in equation (3.8) to obtain the Laplacian representation, $\hat{L}$.

6. Eigen-decomposition of the new matrix $\hat{L}$ to obtain a new set of eigenvalues $\{\lambda_1, \lambda_2, ..., \lambda_m\}$ and eigenvectors $\{\boldsymbol{\phi}_1|\boldsymbol{\phi}_2|...|\boldsymbol{\phi}_m\}$.

7. The top k_{eig} eigenvectors are selected based on the magnitude of the top k_{eig} largest eigenvalues. By selecting only the top k_{eig} eigenvectors, only the significant vectors representing the activity are kept.

8. The eigenvectors $\{\boldsymbol{\phi}_1|\boldsymbol{\phi}_2|...|\boldsymbol{\phi}_{k_{eig}}\}$ are used to represent the human activity in the video sequence.

By selecting only the top k_{eig} eigenvectors during the eigen-decomposition, the k_{eig} eigenvectors should represent only the essential significant activity and serve as a more

general, yet more discriminative model than selecting all possible eigenvectors. Furthermore, it can significantly reduce the dimensionality of the data, and thus reduce the computational complexity.

Classification

For classification purposes, the eigenvectors resulting from the eigen-decomposition step are concatenated by their natural ordering and classified using the k nearest neighbour (kNN) algorithm. We use the Euclidean distance as the distance metric for kNN. The distance between activity eigenvectors is simply the Euclidean distance between each set of eigenvectors

$$D_{ij} = \sqrt{\sum_{m=1}^{k_{eig}} \|\boldsymbol{\phi}_{m,i} - \boldsymbol{\phi}_{m,j}\|^2} \tag{3.10}$$

where k_{eig} is the number of eigenvectors, and $\boldsymbol{\phi}_{m,i}$ is the mth eigenvector for video sequence i.

One important consideration of the kNN algorithm is the selection of the number of neighbours k. The best choice of k depends largely on the data; larger values reduce the effect of noise in the classification but boundaries between the classes become less obvious. Since the range of k is quite small in this context, it is possible to choose k depending on the preliminary results. kNN has two useful properties relevant to our work; firstly, it is non parametric, thus it makes no assumptions about the underlying data - this is useful in this case where no theoretical assumptions are made about the underlying data, e.g. Gaussian mixtures. The second is that kNN is a lazy algorithm; this means that it does not use the training data to generalize and all the training data is kept. This is different from methods such as SVM where some support vectors may be discarded without concern.

3.3 Experimental Results

In this section, we discuss the evaluation and experimental results of the proposed methodology of modelling human activities as graphs. The proposed methodology will be evaluated against the two most common human activity recognition datasets: The Weizmann and KTH datasets.

The Weizmann dataset consists of 90 low-resolution (180×144) video sequences in various scenes performed by 9 actors with 10 natural actions - running, walking, skipping, jumping-jack, jumping forward, jumping in place, gallop side-ways, waving two hands,

waving one hand and bending. Each video sequence is approximately two to three seconds long at 25 frames per second. Since the dataset is small and there is a lack of intra-class training examples available, the leave-one-out cross validation approach will be used for evaluation purposes.

The KTH dataset consists of 2391 video sequences, with approx 500 clips each of six different human activities (walking, jogging, running, boxing, hand waving and hand clapping) performed by 25 actors. Each actor performs an activity in four scenarios: outdoors, outdoors with scale variation, indoors and indoors with scale variation. Each clip is down-sampled to 160×120 and is an average of four seconds in length. The sequences are divided into training and test sets as per the recommendations in [55].

Although both datasets consist of simple staged human actions, the KTH dataset is a more challenging dataset due to the changes in scenes and scale variation. In both datasets, the camera is static. The recognition results will be averaged over 50 runs of the experimental results for both datasets and error bars will be shown on the graphs to show the variability across the 50 runs. The results will be recorded in a confusion matrix. The Weizmann dataset is used as the dataset for the parameter selection, given that the Weizmann dataset is simpler, and both datasets are of roughly the same resolution and contain similar activities.

Feature extraction and description is performed by the Dollar detector and descriptor as described in Section 3.2. The Dollar detector extracts the most significant regions of activity from each the video sequences. As described in Section 3.2, the detector consists of two parameters corresponding roughly to the scale of the extracted regions of significant activity. The two scaling parameters under consideration are σ_s and τ_s from equations (3.2) and (3.3). In our work, the parameters are kept the same as in the original Dollar approach [29] as these values produced the best results on the common activity datasets; therefore $\sigma_s = 3$ and $\tau_s = 2$.

As discussed in Section 3.2, a fixed number of cuboids will be extracted to be able to construct graphs of a consistent size. On the Weizmann dataset, with the default threshold value of $\theta = 2e^{-4}$ for the response function R from equation (3.1), the number of detected regions of activity range from 18 to 600 dependent on the activity. The results in [96] suggest that beyond the 100 most significant regions on such datasets, the recognition results vary very little ($\pm 2\%$). In the typical BoW pipeline, some regions are often disregarded and considered as outliers by the clustering steps. Figure 3.4 shows the

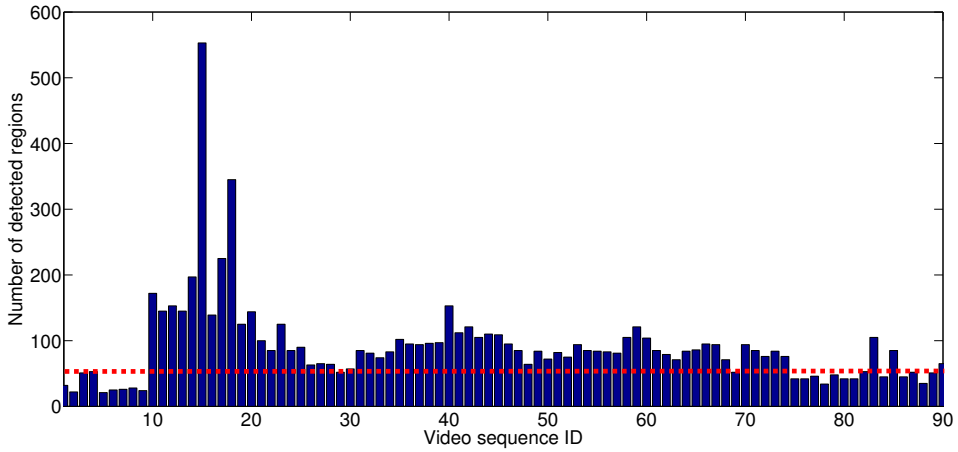


Figure 3.4: Number of extracted cuboids from the Weizmann dataset using the default threshold value of θ.

number of regions extracted with the default threshold, across all the video sequences in the Weizmann dataset. Notably, most of the video sequences contain more than 50 cuboids each while only 20% of the video sequences contain more than 100 cuboids. Considering this, and due to the high variation in the number of regions of activity, only the 50 most significant regions of activity are extracted. Only the 50 most significant regions, the threshold value θ for the response function R from equation (3.1) of the detector must be varied to extract the required 50 regions per video sequence. The variable thresholding algorithm will start at the initial threshold value of $\theta = 2e^{-4}$, and increment the threshold depending on the number of cuboids extracted. The code in algorithm 1 demonstrates the variable thresholding as an algorithm.

Given the completion of the variable thresholding algorithm for each video sequence in the dataset, exactly 50 spatio-temporal interest points are detected for each video sequence. Following this, spatio-temporal cuboids are extracted around the region, as described in Section 3.2. Examples of the response function, cuboid extraction and cuboid regions are shown in Figure 3.5 to Figure 3.8 for different activities from the Weizmann and KTH datasets. Notably, the response function visualised in the figures is stronger for activities showing significant movement, and lower for areas where movement is limited. The extracted cuboid locations, corresponding to the strongest areas of the response function R, show that the human activity is well captured by the interest points. Furthermore, more cuboids are extracted for video sequences containing significant movement. The extracted cuboids are shown as 2D regions which have been extracted by segmenting the cuboid-like

Data: threshold θ, number of regions n

Result: 50 most significant regions of activity

initialization;

while $n < 50$ **do**

  Extract number of regions n using response function R (from equation (3.1)) with threshold θ.;

  Decrease θ by a factor of 1.1;

end

if $n > 50$ **then**

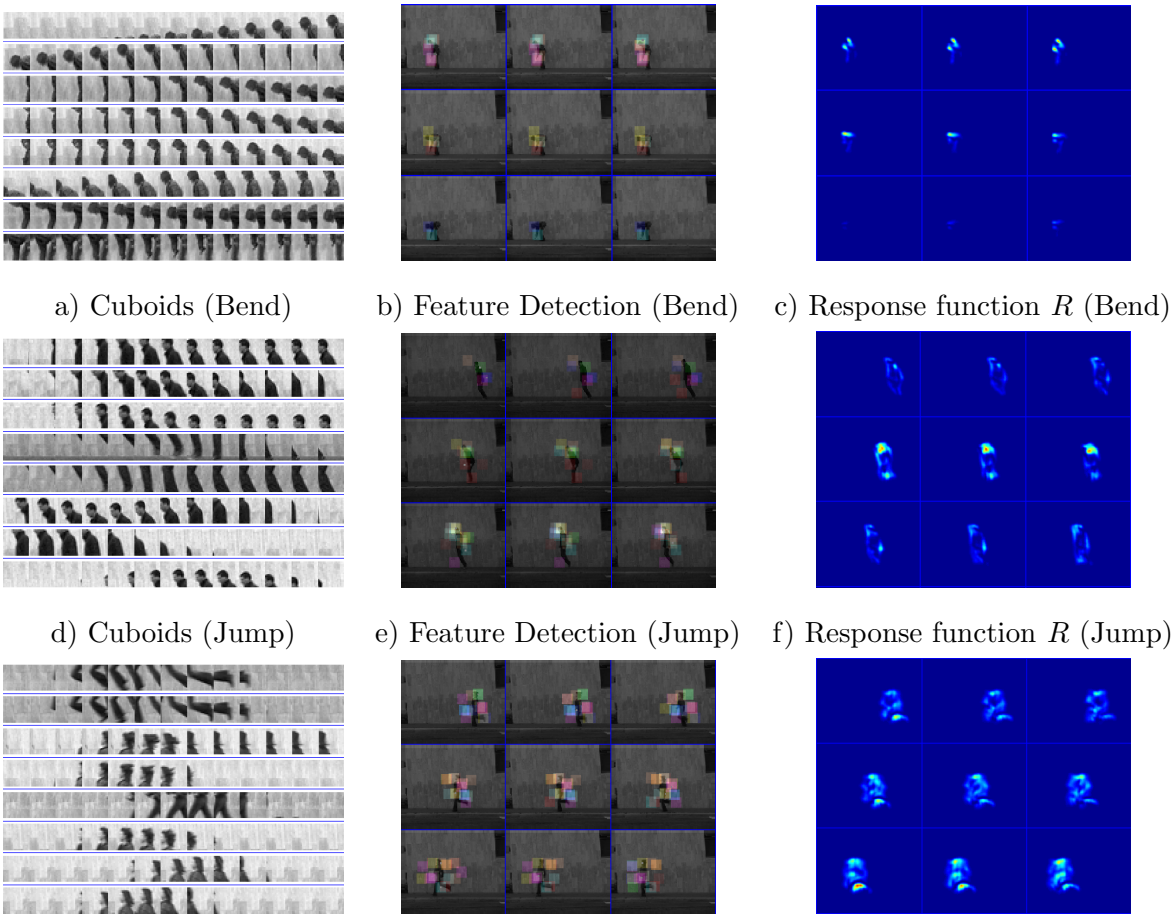
  remove $(n - 50)$ least significant regions of activity to obtain the 50 most significant regions of activity;

end

Algorithm 1: Selecting the 50 most significant regions of activity using variable thresholding.

region temporally. All cuboids in the examples show the activity regions containing the significant activity from each video sequence, evidentiating different activities as shown by the specific movements within their regions. One issue that is apparent from the examples is that when activities are performed quickly, some of the cuboid slices contain no activity as the activity has moved outside the spatial limits of the cuboids. Although this may be a problem for some activities and limit the amount of activity captured, it may help to distinguish between faster and slower activities such as between running and walking.

Following the extraction of the cuboids of significant activity, the feature descriptor must be formed, describing the gradient values representing the spatio-temporal cuboid. As described in Section 3.2, the features are represented by a vector of concatenated gradients over the spatio-temporal region of activity. The feature vector is then reduced in dimensionality using PCA. In the original Dollar paper [29], the length of the vector as a result of the PCA dimensionality reduction is fixed at $k_{pca} = 100$. To determine the appropriate value for k_{pca} for our methodology, we consider the difference in recognition result as k_{pca} is varied. Therefore k_{pca} is varied between 10 and 200, while monitoring the recognition performance on a subset of the original Weizmann dataset. Figure 3.9, shows the recognition error as k_{pca} is varied. It is clear from the figure that the optimum number for k_{pca} is indeed 100. Although the difference in the activity recognition results when considering $k_{pca} = 20$ and $k_{pca} = 100$ is small, the reduction of the descriptor by a factor

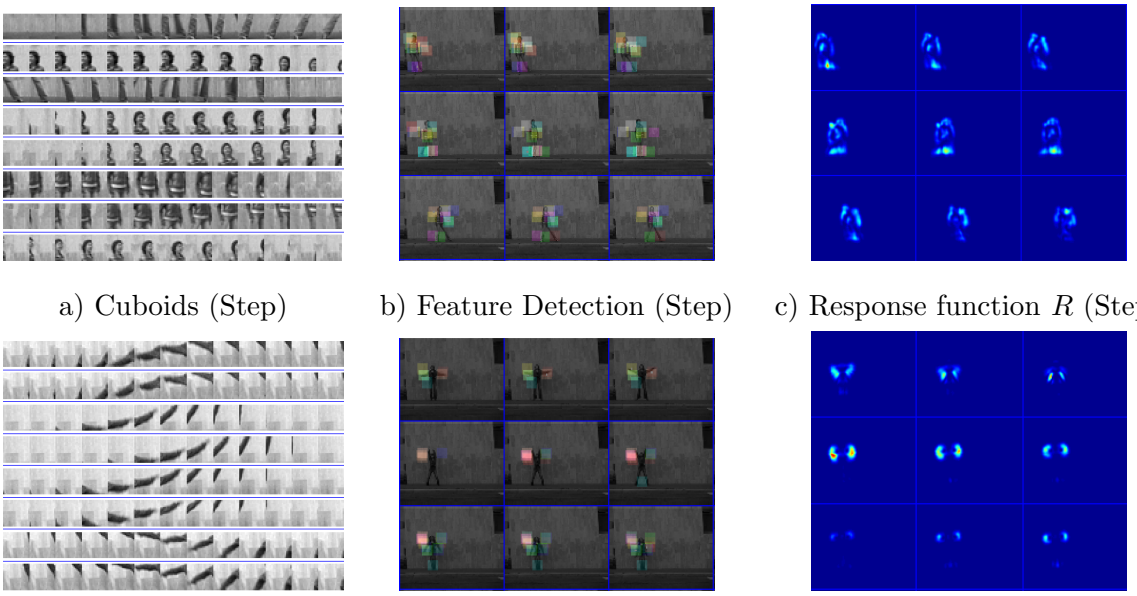


a) Cuboids (Bend) b) Feature Detection (Bend) c) Response function R (Bend)

d) Cuboids (Jump) e) Feature Detection (Jump) f) Response function R (Jump)

g) Cuboids (Run) h) Feature Detection (Run) i) Response function R (Run)

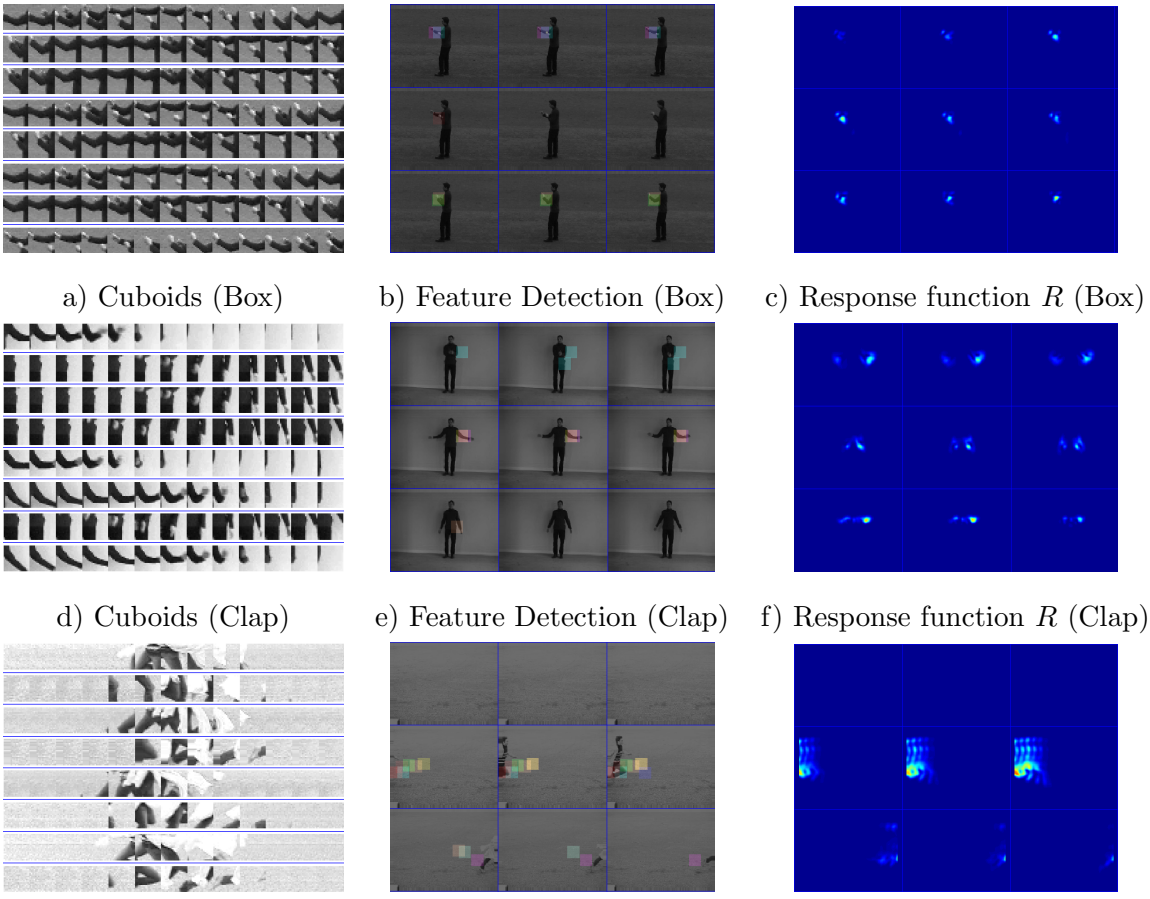
Figure 3.5: Examples of cuboid feature detection and extraction on the Weizmann dataset.



a) Cuboids (Step) b) Feature Detection (Step) c) Response function R (Step)

d) Cuboids (Wave) e) Feature Detection (Wave) f) Response function R (Wave)

Figure 3.6: Examples of cuboid feature detection and extraction on the Weizmann dataset.

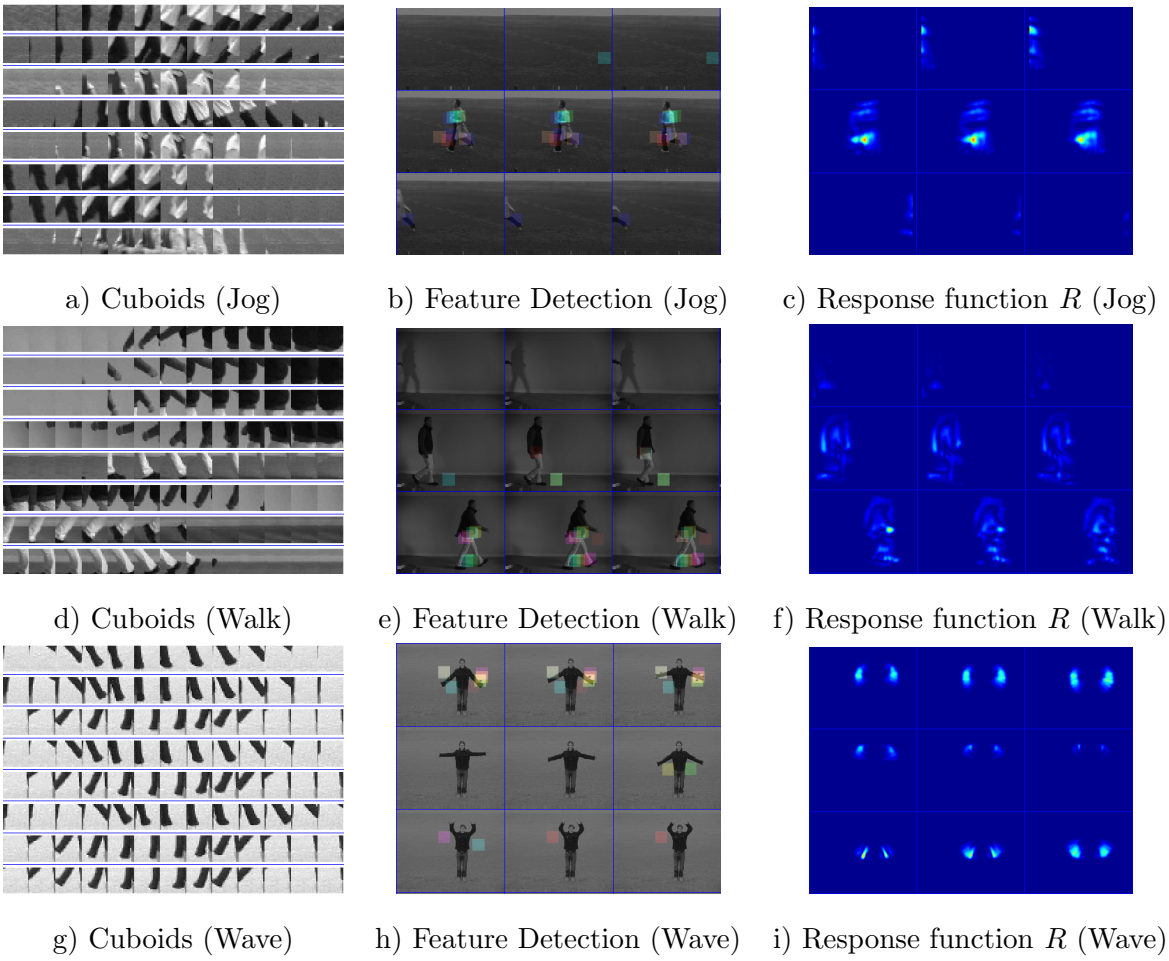


a) Cuboids (Box) b) Feature Detection (Box) c) Response function R (Box)

d) Cuboids (Clap) e) Feature Detection (Clap) f) Response function R (Clap)

g) Cuboids (Run) h) Feature Detection (Run) i) Response function R (Run)

Figure 3.7: Examples of cuboid feature detection and extraction on the KTH dataset.



a) Cuboids (Jog) b) Feature Detection (Jog) c) Response function R (Jog)

d) Cuboids (Walk) e) Feature Detection (Walk) f) Response function R (Walk)

g) Cuboids (Wave) h) Feature Detection (Wave) i) Response function R (Wave)

Figure 3.8: Examples of cuboid feature detection and extraction on the KTH dataset.

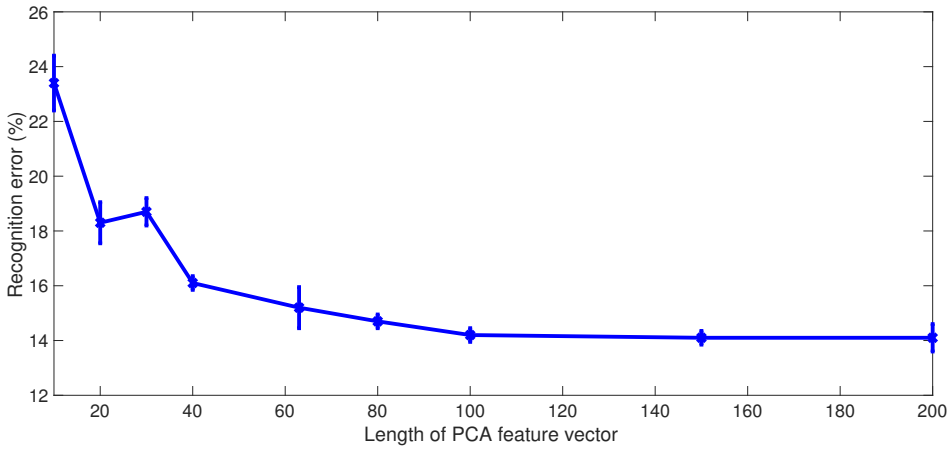


Figure 3.9: Recognition error as the length of the PCA vector k is varied.

of 5 is not significant enough to greatly improve the performance of the proposed graph method. Given $k_{pca} = 100$, each video sequence is therefore represented by 50 features vectors of length 100 each, where each vector summarises the gradient values for each significant region of activity.

Given the 50 feature vectors for each video sequence, the gradient feature similarity matrices can now be constructed as in equation (3.4), described in Section 3.2. An appropriate scaling factor σ must be chosen to construct the similarity matrix. The scaling factor σ from equation (3.4) is empirically chosen such that the activities with high variation between their regions and activities with very little variation between their regions are both well represented by the similarity matrix. The changes in the recognition rates for a subset of Weizmann dataset, when σ is varied between 10 and 100, is shown in in Figure 3.10. It is clear from Figure 3.10 that the lowest recognition errors occur between $\sigma = 50$ and $\sigma = 90$, and the lowest recognition error is obtained when $\sigma = 50$. Considering this, σ is selected as 50. Examples of the similarity matrices for the bend and wave activity are shown in Figure 3.11. Both activities appear reasonably distinct, and some intra-class similarities are visible between the similarity matrices.

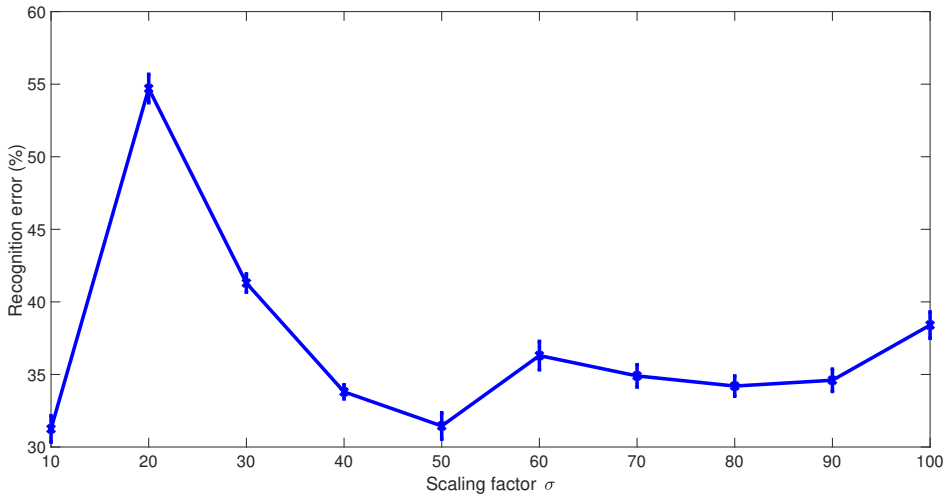
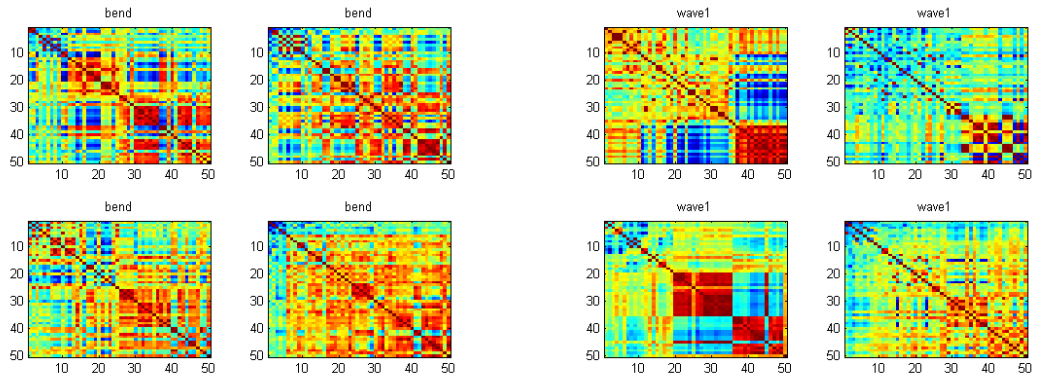


Figure 3.10: Recognition error as the scaling factor σ is varied on a subset of the Weizmann dataset.

As described in Section 3.2, the similarity matrix can be extended to include more discriminant information by including the second localisation term to the model. For matrices created using the additional local information as in equation (3.5), an additional scaling factor is required σ_2 to balance the gradient features and the localisation features.



Bend activity Wave activity

Figure 3.11: Examples of the gradient feature similarity matrices for activities from the Weizmann dataset.

The localisation features considered in this case for the second term are the spatial coordinate vector $\boldsymbol{B} = \{x_i, y_i\}$ and the spatio-temporal coordinate vector $\boldsymbol{B} = \{x_i, y_i, t_i\}$. Each of these vectors refer to the spatial and spatio-temporal location of the cuboid in the video sequence.

Considering the simpler case, where $\boldsymbol{B} = \{x_i, y_i\}$ (spatial case). The possible range of $\|\boldsymbol{B}_i - \boldsymbol{B}_j\|^2$ from equation (3.6) can be derived from the resolution of the video sequence, thus allowing a sensible σ_2 to be chosen.

Considering the case of the Weizmann dataset, with a video resolution of $x_i = 144$, $y_i = 180$, the maximum value resulting from $\|\boldsymbol{B}_i - \boldsymbol{B}_j\|^2$ is $\approx 231^2$, where the minimum value is 0, and the average across all regions of significant activity in the Weizmann dataset is $\approx 67.9^2$. Considering this, the appropriate range of σ_2 for each dataset can easily be determined by its video resolution and frame rate. Given that the cuboids are extracted in order temporally, we consider only including the spatial coordinate differences as the second term. Figure 3.12 highlights the difference in when the second term is used, composed of the spatial coordinate differences, and when σ_2 is varied. From the plot it is quite clear that the optimal value for σ_2 is around 5000. Examples of the gradient and location similarity matrices for the bend and wave activities are show in Figure 3.13. There is a clear similarity present in the bend activity while the similarities in the wave activity aren't as clear.

To provide an even more discriminative model, the local neighbourhood of the significant regions can also be modelled, as described in Section 3.2. The local neighbourhood

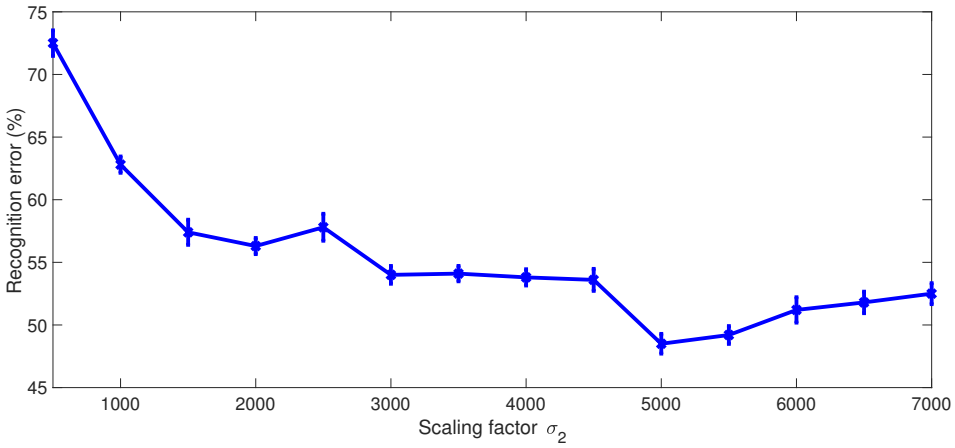
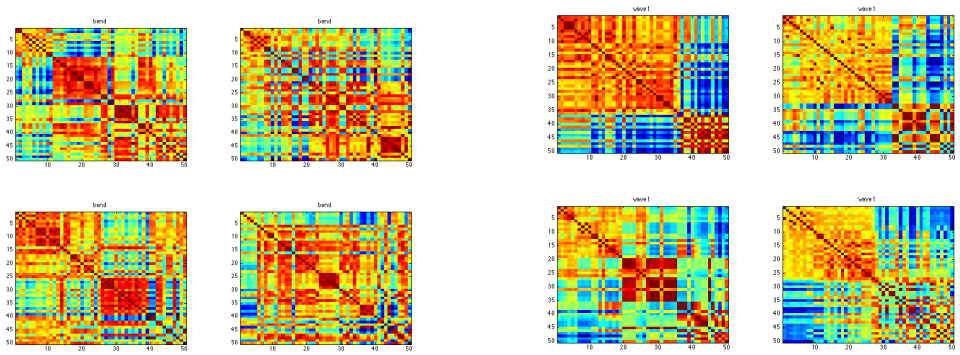


Figure 3.12: Recognition error as the scaling factor σ_2 is varied on a subset of the Weizmann dataset.



Bend activity Wave activity

Figure 3.13: Examples of the gradient and spatio-temporal feature similarity (adjacency) matrices for activities from the Weizmann dataset.

can be modelled as a similarity matrix by either including neighbours which are the nearest significant regions in space-time or the immediate neighbouring (not significant) regions of the same size. The local neighbourhood models the nearest n_{local} neighbours of the region of significant activity as a similarity matrix, using equation (3.4). The nearest neighbours are considered in space-time where the time component is weighted by λ_T as in equation (3.6). λ_T is set to 3 due to this working well in other work on this dataset using the same distance metric. The parameter σ used in constructing the neighbourhood matrix N as in equation (3.6) remains the same as the σ used to construct the main matrix; this is because the main adjacency matrix A contains the same data (gradient values) as the smaller matrix N just on a smaller scale, e.g. 11×11 matrix instead of a 50×50 matrix. Therefore σ is set to 50 for the construction of the neighbourhood matrices. The number of nearest regions (neighbours) n_{local} for the matrix is difficult to choose based on any theoretical basis other than that it cannot be too small, e.g. $n_{local} = 0$ or it cannot be too large $n_{local} = 50$ (all regions included). Considering this, we vary n_{local} and monitor the change in recognition error to determine the appropriate value for n_{local}. Figure 3.14, displays the recognition error as n_{local} is varied, on a subset of the Weizmann dataset. The maximum sensible value of n_{local} is determined as 11, given that beyond this number the recognition error does not improve, and larger values of n_{local} causes computational time problems due to the increase in matrix sizes. From the plot in Figure 3.14, it is clear that the value of n_{local} with the lowest recognition error is 4, thus we choose $n_{local} = 4$.

Examples of the local and immediate neighbourhood similarity matrices for the bend and wave activity are shown in Figure 3.15 and Figure 3.16. For comparison purposes in Figure 3.15 and Figure 3.16, we use matrices of size 11×11 for both local and immediate neighbourhood matrices. In these matrices, the differences between feature classes is not as obvious as for the previous features (gradient and spatio-temporal features).

Next, eigen-decomposition is performed on the graphs. In the case of the gradient term matrix or the gradient and spatial distance matrix, the eigen-decomposition is performed as described in Section 3.2, where eigendecomposition is simply performed on the normalised Laplacian matrix $\hat{\boldsymbol{L}}$ computed from the similarity matrix. The eigen-decomposition of the feature Laplacian matrix results in a set of eigenvectors and eigenvalues representing the activity sequence. One important consideration is the number of top k_{eig} vectors to retain, representing the significant activity. Too few vectors will not provide a discriminant model and be too generalised, while too many vectors will cause over-fitting to each sequence

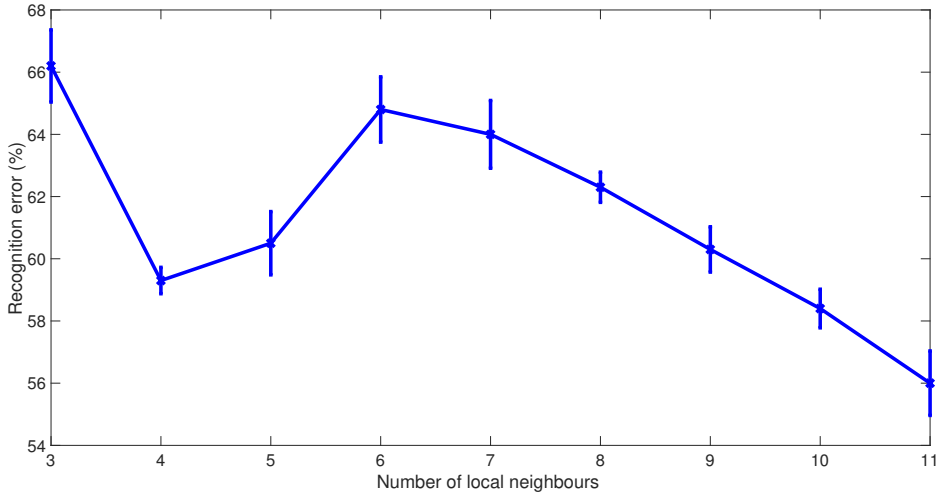
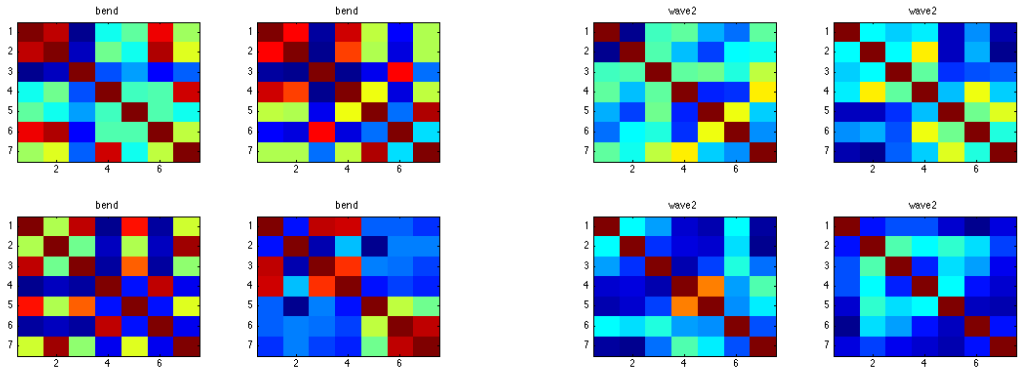
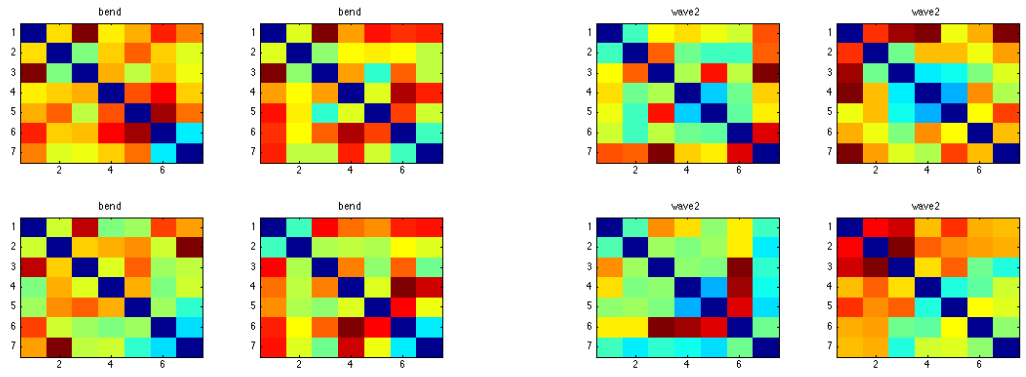


Figure 3.14: Recognition error as the number of local neighbours is varied on a subset of the Weizmann dataset.



Bend activity Wave activity

Figure 3.15: Examples of the local neighbourhood similarity (adjacency) matrices for activities from the Weizmann dataset.



Bend activity Wave activity

Figure 3.16: Examples of the immediate neighbourhood similarity (adjacency) matrices for activities from the Weizmann dataset.

and will not generalise well (while adding to the computational complexity). The value of k_{eig} is chosen by varying the value of k_{eig} and noting the change in recognition error. We vary the value of k_{eig} in the range of 5 to 50 for the standard feature graph, noting the change in recognition error, which is shown in Figure 3.17. It is clear from Figure 3.17 that the best value of k_{eig} is 30, beyond this, the recognition rate does not improve, and computational complexity increases; therefore $k_{eig} = 30$.

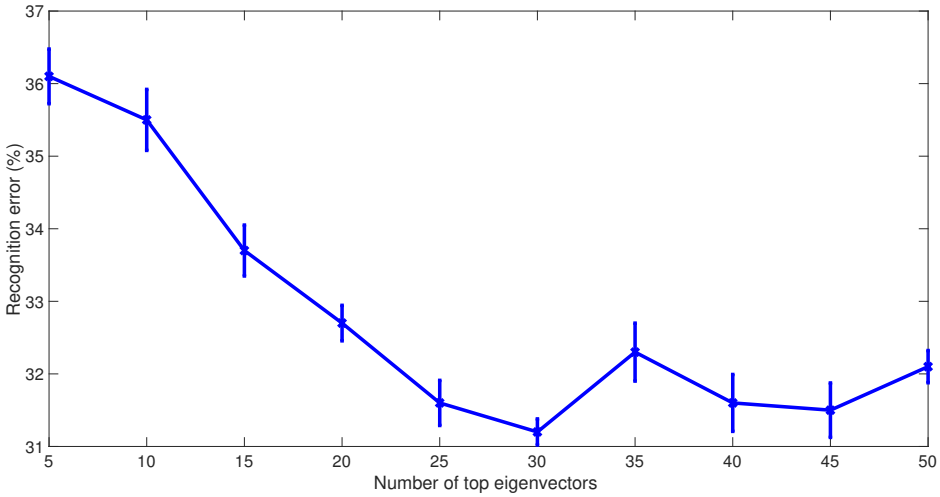


Figure 3.17: Recognition error as the number of top eigenvectors k_{eig} is varied for the feature graph; applied on a subset of the Weizmann dataset.

In the case of the local/immediate neighbourhood graphs, where the surrounding spatio-temporal cuboid are utilised (described in Section 3.2), the eigen-decomposition

steps change, as outlined in Section 3.2. Firstly, eigen-decomposition of the local/immediate neighbourhood Laplacian matrix is performed, before combining with the gradient feature matrix. Similarly to the gradient feature decomposition, we consider the appropriate value for k_{eig}, the number of selected eigenvectors. The local neighbourhood graph is of size 11×11, therefore the eigen-decomposition of the Laplacian yields 11 eigenvectors. The value of k_{eig} is varied between 1 and 11 and the change in recognition error is shown in Figure 3.18. Clearly, the most appropriate value for k_{eig} in this context is 6, therefore $k_{eig} = 6$. Similarly, we consider the appropriate value of k_{eig} for the immediate neighbourhood graph, of size 7×7. Once again, we vary the value of k_{eig} between 1 and 7, and note the change in recognition error, shown in Figure Figure 3.19. Similarly to the local graph, the most appropriate value for k_{eig} from Figure 3.19 6, therefore $k_{eig} = 6$ for the immediate graph representation.

Given the set of eigenvectors representing the local/immediate Laplacian-based graph, the eigenvectors are used as a second term in the construction of the feature matrix, as described in Section 3.2. Similarly, when both the gradient and location terms are used, the representation of the local/immediate cuboids becomes the third term.

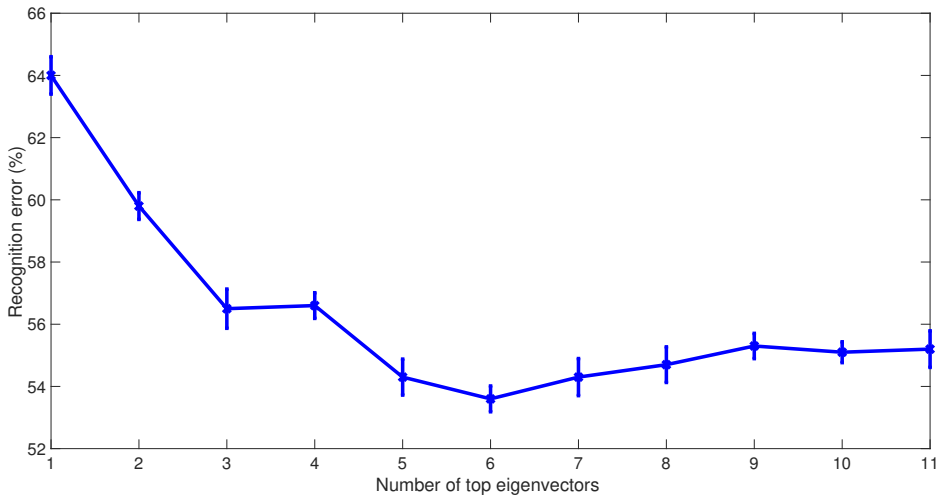


Figure 3.18: Recognition error as the number of top eigenvectors k_{eig} is varied for the local neighbourhood graph; applied on a subset of the Weizmann dataset.

Given the set of significant eigenvectors, representing each activity graph, classification is performed on the sets eigenvectors using kNN, as described in Section 3.2. The kNN algorithm requires selecting the most appropriate value for the number of neighbours k. In order to be consistent across the different activity graph representations and across

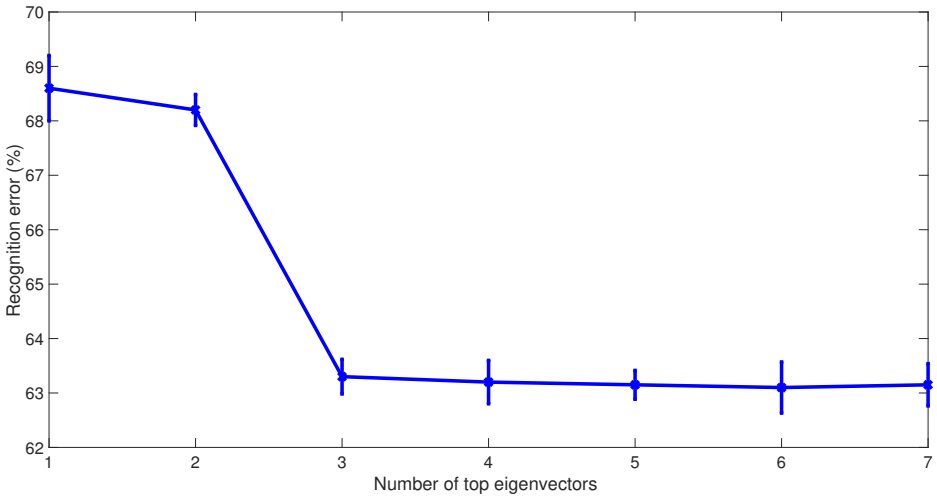


Figure 3.19: Recognition error as the number of top eigenvectors k_{eig} is varied for the immediate neighbourhood graph; applied on a subset of the Weizmann dataset.

different video sequence, the value of k remains constant across all the different activity representations. To determine the most appropriate value for k, we vary the value of k for the gradient graph representation representing a subset of video sequences from the Weizmann dataset. The results of varying k is shown in Figure 3.20. From Figure 3.20, is it clear that the most appropriate value for k is 5, therefore we use $k = 5$ across all experiments.

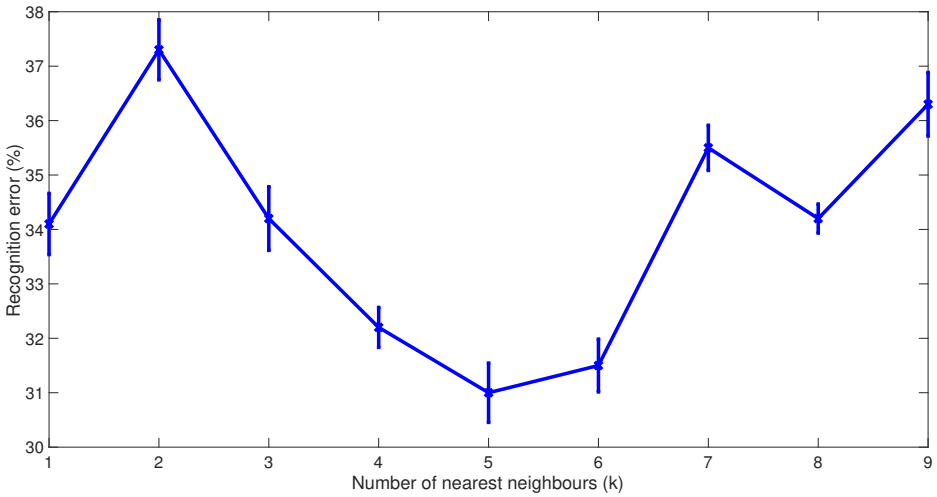


Figure 3.20: Recognition error as the number of nearest neighbours k is varied for the feature graph; applied on a subset of the Weizmann dataset.

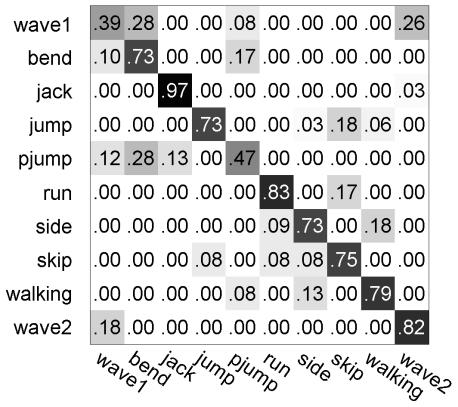
For the recognition tasks on the Weizmann and KTH datasets, we consider combining

Table 3.1: Recognition results on the Weizmann dataset when compared to state of the art approaches.

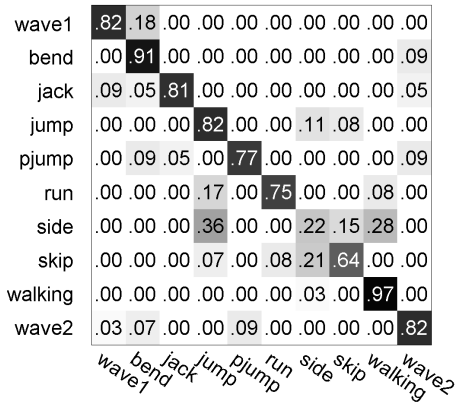
Approach	**Recognition rate(%)**
Gradient feature graph	71.98
Gradient and space-time graph	75.10
Gradient and local neighbourhood graph	59.27
Gradient and immediate neighbourhood graph	59.23
Gradient, space time and local neighbourhood graph	61.25
Gradient, space time and immediate neighbourhood graph	62.18
Original Dollar approach [29]	87.18
Graph based approach using Dollar features [103]	100
State of the art approach based on BoW paradigm [59]	100

different activity graph representations for comparison: gradient feature graph, gradient and space-time graphs, gradient and local/immediate neighbourhood graphs and gradient, space-time, and local/immediate neighbourhood graphs. The recognition results on the Weizmann dataset are evaluated by the recognition rate, using the leave-one sequence out cross validation methodology. This evaluation protocol is consistent with other works [29, 59,103]. The confusion matrices in Figure 3.21, display the recognition rate across different activities, for different feature representation. Notably, the gradient and space-time graphs provide the best recognition results, whilst adding the neighbouring cuboids/regions to the activity model does not improve the results. From these results, it is clear that the space-time information is important for activity recognition and shows a clear improvement in recognition results; meanwhile in this context, the neighbourhood information does not seem to be as useful. Table 3.1 shows the resulting performance of the proposed methodology on the Weizmann dataset, when compared to state of the art approaches. The best result, obtained from the gradient and space-time graphs, does not match the state of the art results, which are 100% on the Weizmann dataset.

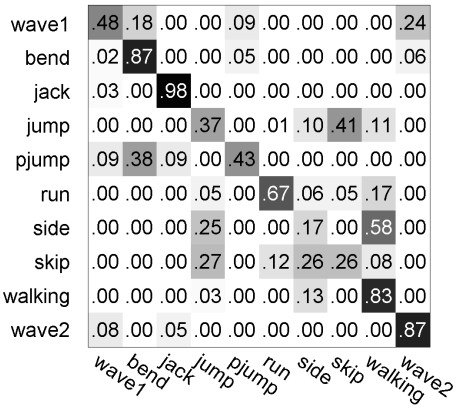
Similarly to the Weizmann dataset, the recognition performance is evaluated on the KTH dataset by using the leave-one sequence out cross validation methodology. The confusion matrices are shown in Figure 3.22, displaying the recognition rate across the different activities. Notably, the gradient and space-time graph outperform the other graph



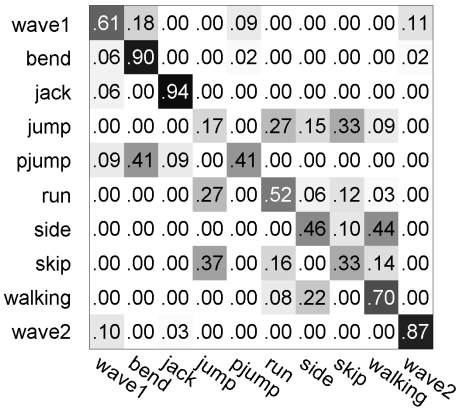
a) Gradient features - 71.98%



b) Gradient and space-time features - 75.10%



c) Local-neighbourhood features - 59.27%



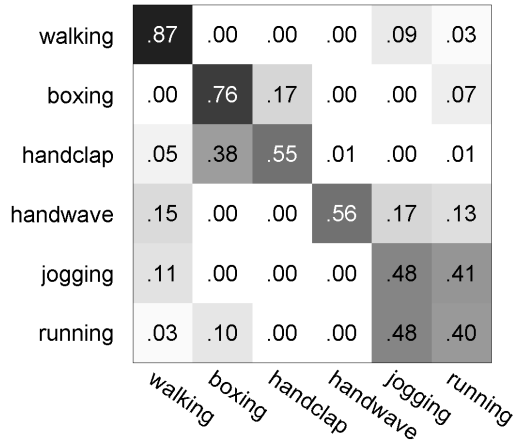
d) Immediate-neighbourhood features - 59.23%

Figure 3.21: Confusion matrices for the recognition results on the Weizmann dataset.

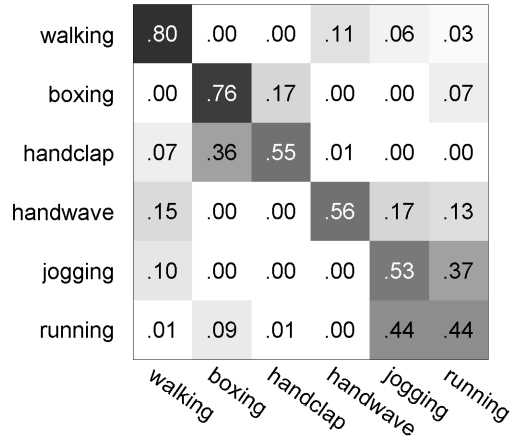
representations. Similarly to the Weizmann dataset, the local and immediate neighbourhood graphs do not aid in the activity recognition performance, while the space-time features improve recognition performance. Table 3.2 shows the resulting performance of the proposed methodology on the KTH dataset, when compared to the state of the art approaches. However, he best recognition result obtained by our methods (66.48%), does not match state of the art methods on this dataset.

a) Gradient features - 66.48%

b) Gradient and space-time features - 64.31%



c) Local-neighbourhood features - 59.70%



d) Immediate-neighbourhood features - 60.07%

Figure 3.22: Confusion matrices for the recognition results on the KTH dataset.

Overall, the recognition performance when compared to the state of the art results is disappointing. However, the majority of state of the art works rely on extracting significantly more space-time features than the 50 in our work, increasing the time-complexity of their methodology significantly. Furthermore, it is suggested that the proposed methodology of modelling relationships between cuboids would be better suited towards more

Table 3.2: Recognition results on the KTH dataset when compared to the state of the art approaches.

Approach	**Recognition rate (%)**
Gradient feature graph	64.31
Gradient and space-time graph	66.48
Gradient and local neighbourhood graph	59.70
Gradient and immediate neighbourhood graph	60.07
Gradient, space time and local neighbourhood graph	62.15
Gradient, space time and immediate neighbourhood graph	62.45
Graph based approach using Dollar features [119]	90.60
State of the art approach based on BoW paradigm [92]	86.83
State of the art approach based on BoW paradigm [51]	95.33

complex activities, for example, interaction recognition or abnormal activities rather than simple human activities, where the contextual information and spatio-temporal relationships are not so significant for modelling the activity.

3.4 Conclusion

In this chapter, a graph based methodology was proposed by modelling the human activity as contextual graphs. In this method, spatio-temporal gradient cuboids were extracted at significant regions of activity, and feature graphs (gradient, space-time, local neighbours, immediate neighbours) were constructed using the similarity matrix. Eigen-decomposition was performed on the Laplacian representation of the similarity matrix, and the most significant eigenvectors were used to classify the human activity using the kNN algorithm. The proposed methodology was evaluated on the Weizmann and KTH human activity datasets, although the results did not match those of the state of the art. However, in this approach, the time-complexity of the methodology is reduced considerably compared to others. Furthermore, this method models the connectivity between activity cuboids, whereas the current state of the art approaches generally rely on clustering cuboids, and do not consider contextual relationships between features. Given this, we suggest that our approach may be better suited at detecting more complex activities, such as human interactions, abnormal activities and contextual group activities.

Chapter 4

Anomalous Activity Detection

4.1 Introduction

While simple human activity is still useful for basic recognition tasks, a more complete solution is required for activity analysis in complex real world scenes. Due to this, the attention of research has moved on from simple human activities to detecting activities in real-world crowded scenes. In particular, towards approaches which focus on the detection and identification of uncharacteristic activities for a particular scene [50, 58, 105]. Such attention has been brought towards anomalous human activity recognition due to the variety of potential real-world applications, especially in the areas of surveillance and security.

The focus of this research will be on the online detection of abnormal human activities. More specifically, being able to detect abnormal behaviours in a video sequence given the known expected normal behaviour for the scene. Given a certain context, there is a notion of what is considered to be normal human activity, and conversely, abnormal activity. The concept of an activity being defined as abnormal is heavily dependent on its context; for example, a pedestrian walking down a high-street would be considered normal, but a pedestrian walking across a busy motorway would be considered abnormal. In such a detection system, the general (normal) behaviour is learnt, that is the typical observable actions of persons or other moving objects in the scene. The anomalies can thus be defined as the interesting (often uncommon) behaviour, in other words, events that do not conform to the learnt patterns. Usually, abnormal activities occur with low probability with respect to the probability of detecting normal trained activity. The area of detecting abnormal activity from video sequences is well researched in computer vision, with a wide

variety of proposed methods. In simple human activity recognition, the main recognition pipeline began with low-level feature extraction, followed by some feature representation (such as bag of words), followed by basic classification. In complex, crowded scenes, the general low-level approaches to feature representation are unreliable and the performance of such methods tend to degrade due to factors such as scene clutter, occlusions and general density of unsteady flow in the scene.

One representation of activity modelling is based on longer-term object trajectories. Modelling activities as object trajectories is usually done either by explicitly or implicitly segmenting and tracking each object in the scene, and fitting models to the resulting tracks [25, 45, 100]. Whilst these methods are reliable for open, uncluttered sparsely-populated areas, they perform poorly in crowded scenes, especially when such pedestrians are often occluded by other people, objects or by moving vehicles. A number of shorter-term tracking methods have been proposed, such as modelling the motion as histograms of optical flow [2,126] or as a mixture of probabilistic principal component analysis (PCA) models [50]. Notably, a number of medium-term tracking methods have also been proposed such as streaklines [65], which aim to model the medium-term flow of movement in crowded scenes, without relying on long term tracks.

Recently, several notable methods for abnormal activity detection have been proposed. [58] proposed a detector that accounts for both appearance and dynamics using a set of mixture of dynamic texture models. Li *et al.* [58] also introduced a dataset of densely crowded pedestrian walkways which consists of non-staged, realistic anomalies such as bicyclists and electric-vehicles. Thida *et al.* [105] proposed to model the optical flow using a spatio-temporal Laplacian eigenmap to extract different crowd activities from videos. The motion patterns are clustered using k-means on the graph in the embedded space and a multivariate Gaussian mixture model (GMM) is used to represent the regular motion patterns. Basharat *et al.* [7] modelled the object motion patterns using a probability density function (pdf) at each pixel to extract the speed and size of the tracks, then unsupervised Expectation Maximisation (EM) was used to learn the tracks of every GMM. Their proposed method successfully detected both local and global anomalies. Kratz *et al.* [52] also modelled the motion patterns using GMMs except using gradients as a 3D distribution instead of optical flow. A dictionary of activity prototypes was learnt by identifying statistically similar cuboids using the KL-divergence between probabilistic models. Finally, GMM based Markov random fields (GMM-MRF) were used in [72] for

abnormal activity detection.

A new online activity monitoring approach is adopted in this research based on forming a dictionary of activities extracted by analysing the video information from a training set and assessing a new activity using detection theory. During the training stage, the movement in the scene is estimated by streaklines [65]. The trajectory based modelling using streaklines [65] is used for localising and characterising human activity. Each distinct moving region is then characterised statistically using GMMs by its motion and location parameters, forming a dictionary of normal activities for the given scene. New activities are detected in a second stage, where the scene is observed and all activities are extracted and tested against the existing dictionary of activities. Any new human activity is compared statistically using Kullback-Leibler (KL) divergence with the distributions of all learnt activities from the dictionary. If the activity corresponds to one of those already recorded in the dictionary, according to a threshold on the KL divergence, then its parametric representation from the dictionary is updated accordingly. Otherwise, an alarm can be triggered and a new human activity is added to the dictionary.

The rest of the chapter will be organised as follows: Section 4.2 provides an overview of the proposed anomaly detection methodology. Section 4.3 describes the proposed approach to movement estimation, including streakline modelling. Section 4.4 describes the activity representation via mixture of Gaussians. Section 4.5 describes the anomalous activity detection stage while Section 4.6 describes the localisation of activities. Section 4.7 describes the experimental results and finally the conclusions are provided in Section 4.8.

4.2 Proposed Anomaly Detection Methodology

The proposed method is an online system designed to distinguish between normal and abnormal behaviours in real-world video sequences. The proposed method could be applied to any real-world video sequence, and the method should be able to distinguish between normal and abnormal human behaviours in the given scene. The processing stages of the proposed method are illustrated in Figure 4.1.

A more detailed overview of the method is provided below:

1. **Video sequences input** - The video sequence is provided as input to the proposed method, defining what is considered normal in the training, and the unknown behaviour in the testing.

2. **Motion estimation** - Block matching or streaklines are used to estimate the motion over several frames in the video sequences.

3. **Histograms of motion flows** - The motion will be segmented, firstly using a simple algorithm to label connected components. Then, the number of distinct regions is estimated by the number of peaks from the local or global histograms of flow vectors.

4. **Motion segmentation** - Using the number of peaks from the local/global histograms of flow as the number of components, the expectation maximization (EM) algorithm is applied to each region under the Gaussian modelling assumption. The subsequent new regions are labelled accordingly. Each distinctly segmented region in the previous step has its movement represented by a multi-variate Gaussian mixture model.

5. **Activity representation by GMMs** - A dictionary of normal activities will be constructed by using the KL divergence as a statistical measure of similarity between the GMMs for the regions in the video sequence. KL Divergence will be computed between the regions in the image and the dictionary of activities. If the divergence is above a certain threshold, a new activity will be created in the dictionary. At each iteration, if new regions are detected which consist of activities already present in the dictionary, the parameters of the existing model (activity) will be updated.

6. **Detecting anomalous activities** - When new activities are detected in the test set (anomalies), the frame number and spatial coordinates (location) will be recorded. The frame number and location will be compared to the ground truth data to provide numerical evaluation results.

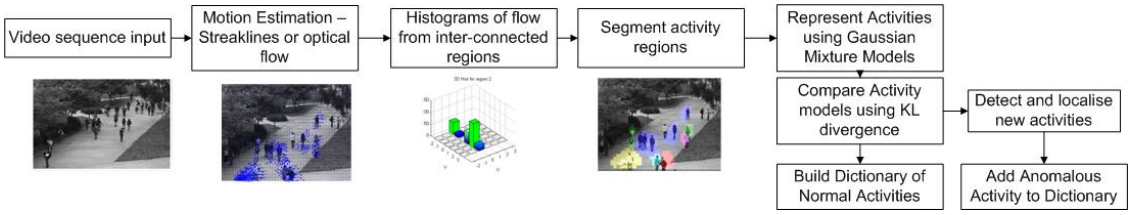


Figure 4.1: Processing blocks for the proposed method of anomalous activity recognition.

4.3 Movement Estimation

The first stage of the proposed approach consists of movement estimation. In our approach, frames from the video sequence are divided into spatio-temporal blocks of a certain size. We propose two methods of block-based movement estimation: optical flow estimation using block matching and a medium-term movement estimation method called streaklines. We begin by discussing the optical flow approach, followed by some considerations and issues which lead to a streaklines based approach. To begin, the local movement is estimated for each pixel block, between two frames by using a generic optical flow estimation method, for example the well known block matching algorithm, which is used extensively in video coding. Block matching is a simple, yet effective motion estimation method which is very commonly used in video compression [68]. Block matching attempts to find the motion of image patterns corresponding to objects between two subsequent frames. More specifically, each image frame is spatially divided into regions, often called macro blocks. Each macro block in the current frame is compared to its corresponding block in the subsequent frame and its adjacent neighbours. After finding the best correlated block of pixels from the subsequent frame, the difference between the coordinates of the pair of blocks gives the displacement vector. This vector is associated to the movement of the area in the scene corresponding to the given macro block of pixels. This process is repeated for all macro blocks in the image. The search area for each macro block is an important consideration and is dependant on the maximum amount of movement expected in each macro block. Faster motion requires a large search window, but consequently as the search window increases, the process becomes increasingly computationally expensive. The size of the macro block is also an important consideration; too large and small motion patterns are lost, too small and the computationally complexity required increases significantly, as does the amount of noise. The matching of a macro block with another is done based on the result of a cost function. The macro block that results in the least cost is typically the one that matches closest to the current block. Various cost functions exist, the most popular, and the one chosen for this application is Mean Squared Error (MSE).

$$\text{MSE} = \frac{1}{M} \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} (I_{ij}^{t} - I_{ij}^{t+l})^2 \tag{4.1}$$

where $M \times M$ is the size of the macro block and I_{ij}^{t} and I_{ij}^{t+l} are the pixels i, j being compared in the reference macro block for frame t to the macro block in the current frame

$t + l$, where l is the number of frames skipped. Therefore a vector is obtained for each macro block in the frame, where each vector stipulates the movement of the macro block from the current frame to its subsequent reference frame. The set of motion vectors for the whole frame is used to represent the motion between the reference frame and the current frame. This process is repeated for all frames in the video sequence.

Modelling of Streaklines

One issue that arises from using optical flow alone is the difficulty in capturing unsteady movement in crowded scenes. To alleviate this problem, we propose the use of streaklines [65]. Streaklines correspond to tracking fluid particles that have passed through a particular location in the past. Streaklines provides a solution to non-smooth movement based on a Lagrangian framework for fluid dynamics [65].

In fluid mechanics there are different vector field representations of flow:

- **Streamlines**, which are tangent to the velocity vectors at every point in the flow. These correspond to traditional optical flow.
- **Pathlines**, which are trajectories that individual particles in a fluid flow follow. These directly correspond to integration of optical flow over time.
- **Streaklines**, which represent the locations of all particles at a given time that passed through a particular point.

For flows that are steady and unchanging, these three representations provide similar results. When the flows are unsteady and changing over time, they are notably different. First of all, in crowded dense scenes, streamlines (and similarly optical flow) will leave spatial gaps in the flow and provide choppy transitions over time. Hence, it would not provide smooth fluid-like flow for crowded videos. Pathlines overcome this problem by filling the spatial gaps, but do not allow for detection of local spatial changes which is critical in activity recognition based tasks. Furthermore, pathlines require $L \times (L-1)/2$ particles for L pathlines while streaklines only require L particles for the equivalent flow representation. Since streaklines make use of a Lagrangian model for fluid flow, much of these problems are alleviated; such a model is ideal to exploit the dynamic changes in crowded scenes (where frequent changes in the flow are expected) whilst also filling spatial gaps and providing a smooth transition of flow over time.

To compute the streaklines, dense optical flow is computed from frame to frame using the block matching method described above. Streaklines can be computed by initializing a set of particles at every time instance in the field and propagating them over time and in space using the optical flow field. This results in a set of paths, each belonging to one point of initialization. To explain how streaklines are calculated let $x_i^p(t), y_i^p(t)$ be particle at time t, initialised at point p and frame i for $i, t = 0, 1, 2, ..., T$. Then, repeated initialisation at p implies $(x_i^p(i), y_i^p(i)) = (x_0^p(0), y_0^p(0))$. Particle advection is achieved by

$$\begin{aligned} x_i^p(t+1) &= x_i^p(t) + u(x_i^p(t), y_i^p(t), t) \\ y_i^p(t+1) &= y_i^p(t) + v(x_i^p(t), y_i^p(t), t) \end{aligned} \tag{4.2}$$

where u and v represent the velocity field of the optical flow. This produces a series of curves, all starting at point p and tracing the path of the flow from that point in frame i. For steady flow all these curves lie along the same path, but for unsteady flows the curves vary in direction and shape, characteristic of pedestrian flow in crowded scenes. This setup allows streaklines to propagate velocities, given by the instantaneous optical flow $\Omega = (u, v)^T$ at the time of initialization, along the flow like a material. To this end, we can then define an extended particle i as a set of position and initial velocity

$$P_i = \{x_i(t), y_i(t), u_i, v_i\} \tag{4.3}$$

where $u_i = u(x_i^p(i), y_i^p(i), i)$ and $v_i = v(x_i^p(i), y_i^p(i), i)$.

Similar streaklines will correspond to similar trajectories of particles from neighbouring pixels. Unlike in [65] where streaklines are computed for each pixel, we associate each streakline with a block of pixels of a fixed size by computing the marginal median as the streakline estimate for each block of pixels.

We consider two different ways to represent the streaklines: single vector representation and multi-vector representation. In the case of single vector representation, we apply PCA on the streakline vectors in order to extract the principal eigenvector indicating the direction of movement for each pixel-block. In the multi-vector streaklines approach, we have several movement vectors which are in a smooth sequence. Therefore, for the single-vector streaklines, we estimate a single movement vector for a block of pixels, spanning several frames. For the multi-vector streaklines, we consider modelling the orientation and magnitude independently. We consider defining a more intuitive space, in the polar coordinate space, characterizing the orientation and intensity of local movement instead

of the Cartesian coordinate space. For the multi-vector streaklines, we compose a feature vector consisting of several orientation features and a single magnitude feature, spanning several frames. It is expected that the multi-vector streaklines may perform better at characterising individual movements in the scene, but may capture frivolous movements that are not important characteristic feature of the activity at hand. On the other hand, the single-vector streaklines may better define simple movements over several frames, despite potentially losing smaller human movements.

4.4 Activity Representation Using Mixtures of Gaussians

Given the streaklines representing each spatio-temporal block of pixels, the video sequence is segmented into distinct moving regions. To begin, the motion is segmented into inter-connected regions, considering 4 connected neighbouring blocks of pixels. The inter-connected regions will be further segmented into distinct moving regions, each characterised by multi-variate Gaussian mixture models, representing streaklines. GMMs have widely been used as a parametric model of motion [7, 105]. A Gaussian mixture model Θ is a weighted sum of K component Gaussian densities as given by

$$p(x|\Theta) = \sum_{k=1}^{K} w_k p_k(x|\mu_k, \Sigma_k) \tag{4.4}$$

where x is a d dimensional streakline vector obtained from a set of streakline vectors $D = \{x_i, ..., x_N\}$ and N is the number of streakline vectors. w_k is the mixture weight and each p_k is the Gaussian density for component k.

We consider in the following a multivariate Gaussian function for modelling the streaklines characterising a compactly moving region. Each component k is therefore a multivariate Gaussian density given by

$$p_k\left(x|\theta_k\right) = \frac{1}{(2\pi)^{d/2}\left|\Sigma_k\right|^{1/2}} e^{-\frac{1}{2}(x-\mu_k)^t \Sigma_k^{-1}(x-\mu_k)} \tag{4.5}$$

with parameters $\theta_k = \{\mu_k, \Sigma_k\}$ (mean and variance).

The complete GMM is parametrized by the mean vectors, covariance matrices and mixture weights from all component densities. Each component density will represent a specifically moving region in the video frame. Each distinct region in the frame will be represented by the means and variances of the streaklines.

We adopt two different approaches of movement segmentation for defining activities. The first approach consists of global segmentation, where each GMM component corresponds to a certain movement, defined irrespective of their location in the frame. This means that several regions of movement from the frame may correspond to the same activity. For example, a pedestrian walking to the right in one part of the scene may correspond to the same GMM component as a pedestrian walking to the right on the other side of the scene. In the second approach, we adopt a local approach, where each region of movement is defined locally. In this case, each interconnected region is considered a multivariate GMM, where the GMM parameters are estimated from the streakline data. Each GMM component corresponds only to the segmented regions inside that interconnected region. While the global approach will produce fewer but spurious regions of activity, the second approach will produce additional regions of movement, each of them compactly defined in the space of the video frame.

The inter-connected regions are segmented using the label results obtained from applying the iterative Expectation-Maximization (EM) algorithm, under the GMM assumption. Histograms of movement flow are generated for each inter-connected region (local) or for the whole scene (global). The EM algorithm is initialised by using the set of initial parameters obtained from the histogram peaks (simply their values, an approximation of the modes). The algorithm then iteratively updates the parameters (by repeating the E and M steps), until convergence. The E and M step can be computed as follows:

E Step: Compute the membership weights by:

$$w_{ik} = p(i|x_i, \theta) = \frac{p_k(x_i|z_k, \theta_k)\alpha_m}{\sum_{m=1}^{K} p_m(x_i|z_m, \theta_m)\alpha_m} \tag{4.6}$$

where x_i is the motion vector for region i (component i), where the points lie in cluster k and given that: $1 \leq k \leq K, 1 \leq i \leq N$. And K is the number of mixture components. $z = \{z_1, ..., z_K\}$ is a vector of K binary indicator variables. z is a random variable representing the identity of the mixture component that generated x. α_m is the mixture weight for component m. The membership weights reflect the uncertainty of vector x_i and parameters θ, about which of the K components that generated x_i.

The weights are computed for all data points, and all mixture components using the equation above.

M Step: The membership weights are now used to calculate new parameter values, where the new mixture weight is:

$$\alpha_k^{new} = \frac{\sum_{i=1}^{N} w_{ik}}{N} \tag{4.7}$$

And the new parameters become:

$$\mu_k^{new} = \frac{\sum_{i=1}^{N} w_{ik} x_i}{\sum_{i=1}^{N} w_{ik}} \tag{4.8}$$

and

$$\Sigma_k^{new} = \frac{\sum_{i=1}^{N} w_{ik} \cdot (x_i - \mu_k^{new})(x_i - \mu_k^{new})^t}{\sum_{i=1}^{N} w_{ik}} \tag{4.9}$$

After the new parameters have been computed, the M step is complete and the next iteration can begin.

The algorithm ends when convergence is reached, or more specifically this is when the log-likelihood computed after each iteration is no longer changing in a significant manner (from one iteration to the next). The log-likelihood is defined as follows:

$$\log l(\theta) = \sum_{i=1}^{N} \log p(x_i|\theta) = \sum_{i=1}^{N} (\log \sum_{k=1}^{K} \alpha_k p_k(x_i|z_k, \theta_k)) \tag{4.10}$$

where the complete set of parameters for the Gaussian function is given by $\theta = \{\alpha_1, ..., \alpha_K, \theta_1, ..., \theta_K\}$ and $p_k(x_i|z_k, \theta_k)$ is the Gaussian density for component k.

Each distinct moving region is therefore represented by the parameters of the respective Gaussian distribution. The segmentation is repeated across the entire video sequence for either the local or global segmentation approach, leading to a set of streakline GMM models characterising the movement in the scene.

4.5 Activity Detection using Statistical Relevance Criterion

The model described above leads to creating a set of streakline GMM models for the entire scene. Each of these GMM models correspond to one component of the GMM model and can be characterized statistically by its streakline statistics, corresponding to its mean vector $\boldsymbol{\mu}_i$ and covariance matrix $\boldsymbol{\Sigma}_i$. Different activities will be detected by comparing the GMM statistics using a statistical relevance criterion. For this purpose, we use the Kullback Leibler (KL) divergence. KL divergence is a non-symmetric measure of the difference between two probability distributions P and Q. The paper [53] describes the general case in more detail. In the context of this work, the pdfs representing regions of movement in the scene are modelled using GMMs. The KL divergence will indicate the

changes from one region of movement to another in the scene, by measuring the statistical similarity of differences in the movement. Significant differences in the KL divergence are evidence of significant differences in the movement of various regions of the scene which could suggest hard interactions, while smaller differences could indicate similarity in the movement. However, the decision depends on the location of such moving regions with respect to each other and the KL divergence is applied in this context as well.

Similarly, we could consider using the Jensen-Shannon Divergence, obtained from the KL divergence. The Jensen-Shannon divergence has some notable differences, including that it is already symmetric and it is always a finite value.

The KL divergence can be written as $\mathrm{KL}(f_1, f_2)$ for densities f_1 and f_2, and in general form is given by

$$\mathrm{KL}(f_1, f_2) = \int f_1(x) \log \frac{f_1(x)}{f_2(x)} \mathrm{d}x \tag{4.11}$$

For most densities, $\mathrm{KL}(f_1, f_2)$ is not available in closed form and needs to computed numerically; one exception to this is if both densities are Gaussian distributions. KL divergence for GMMs was used in human abnormality detection to distinguish between normal and anomalous regions [52] and more generally to compute difference between probabilistic models. In the research presented in this chapter, it will be used to compute the difference between the Gaussian density functions of two regions in the video sequence.

The KL divergence between activity region $\mathcal{A}_i$ with mean vector μ_i and diagonal covariance matrix $\mathbf{\Sigma}_i$ and activity region $\mathcal{A}_j$ with mean vector $\boldsymbol{\mu}_j$ and diagonal covariance matrix $\mathbf{\Sigma}_j$ is given by [30]:

$$\mathrm{D}_{\mathrm{KL}}(\mathcal{A}_i || \mathcal{A}_j) = 0.5[\log(\det(\mathbf{\Sigma}_j)/\det(\mathbf{\Sigma}_i)) + \mathrm{tr}(\mathbf{\Sigma}_j^{-1}\mathbf{\Sigma}_i) + (\boldsymbol{\mu}_j - \boldsymbol{\mu}_i)'\mathbf{\Sigma}_j^{-1}(\boldsymbol{\mu}_j - \boldsymbol{\mu}_i) - d] \tag{4.12}$$

where d is the number of dimensions (i.e. dimension of the streakline vectors). From observation of equation (4.12), one issue that may arise is when matrix Σ_I is singular and consequently its determinant is zero, so division by zero occurs. This can be alleviated by simply setting $\mathbf{\Sigma}_i$ to a very small value when $\mathbf{\Sigma}_i$ is 0.

Equation 4.12 therefore provides the difference between the two probability distributions for streakflow models $\mathcal{A}_{I(t)}$ and $\mathcal{A}_{J(t)}$.

One downside of using the standard KL divergence from Equation 4.12 is that it is not symmetric. A symmetrised version of the KL divergence can be computed by:

$$\mathrm{D_{SKL}}(\mathcal{A}_i||\mathcal{A}_j) = \frac{1}{2}\left[D_{KL}(\mathcal{A}_i||\mathcal{A}_j) + D_{KL}(\mathcal{A}_j||\mathcal{A}_i)\right] \tag{4.13}$$

Therefore, equation (4.13) will provide the difference between the two probability distributions for activities $\mathcal{A}_i$ and $\mathcal{A}_j$, and a smaller value will indicate similarity in the activities being observed.

Similarly, we can compute the Jensen-Shannon divergence for comparison by:

$$\mathrm{D_{JSD}}(\mathcal{A}_i||\mathcal{A}_j) = \frac{1}{2}\left[D_{KL}(\mathcal{A}_i||M) + D_{KL}(\mathcal{A}_j||M)\right] \tag{4.14}$$

where $M = \frac{1}{2}[\mathcal{A}_i + \mathcal{A}_j]$.
Equation (4.14) will be used later for comparing the performance of the symmetric KL divergence to the Jensen-Shannon Divergence.

To begin the detection algorithm, we calculate the streakline Kullback-Leibler divergence (SKL), according to equation (4.13), between the streakline distributions corresponding to all pairs of moving regions identified in the scene.

Given such computations between streakline distributions, a new activity is decided when we have:

$$\mathrm{KL}(\mathcal{A}_k, \mathcal{A}_j) > \Theta \tag{4.15}$$

for $k, j = 1 \ldots, N$, and where Θ is a threshold characterizing the novelty in the scene. If equation (4.15) is fulfilled, then a new activity $\mathcal{A}_k$ is added to those recorded in the dictionary of activities characterizing the scene. If equation 4.15 is not fulfilled then we would have $\mathcal{A}_k \equiv \mathcal{A}_j$ and the observed activity corresponds to one of the activities currently recorded for that scene. In this case the parameters $\boldsymbol{\mu}_j$ and $\boldsymbol{\Sigma}_j$ corresponding to the activity $\mathcal{A}_j$ are updated. The parameters $\boldsymbol{\mu}_j$ and $\boldsymbol{\Sigma}_j$ corresponding to the activity $\mathcal{A}_j$ can be updated as follows:

$$\boldsymbol{\mu}_j^{new} = \frac{\boldsymbol{\mu}_j N_j + \boldsymbol{\mu}_k N_k}{N_j + N_k} \tag{4.16}$$

$$\boldsymbol{\Sigma}_j^{new} = \frac{\boldsymbol{\Sigma}_j N_j + \boldsymbol{\Sigma}_k N_k}{N_j + N_k} \tag{4.17}$$

where $\boldsymbol{\mu}_j^{new}$ and $\boldsymbol{\Sigma}_j^{new}$ are the new parameters for activity $\mathcal{A}_j$, and $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ are the parameter of the new activity in the scene. N_j and N_i correspond to the total number of blocks present for the activity.

In the case of multi-vector streaklines, the KL divergence is split into two terms, one for the orientation of local motion and another for its intensity. The magnitude of the movement is indicated by the eigenvector corresponding to the highest eigenvalue, after applying the PCA onto the vectors composing the streakline. Meanwhile, we consider the orientation angles for all the displacement vectors which make up the streakline. Let $\mathcal{A}_{o,k}$ be the orientation model for Activity k and $\mathcal{A}_{m,k}$ be the magnitude model for Activity k. Likewise, let $\mathcal{A}_{o,j}$ be the orientation model for Activity j and $\mathcal{A}_{m,j}$ be the magnitude model for Activity j. Then we define a new activity criterion:

$$k_o \mathrm{KL}(\mathcal{A}_{o,k}, \mathcal{A}_{o,j}) + k_m \mathrm{KL}(\mathcal{A}_{m,k}, \mathcal{A}_{m,j}) > \Theta_s \tag{4.18}$$

for $j = 1 \ldots, N$, where k_o and k_m are weights in the range [0,1] for the orientation and magnitude respectively such that $k_o + k_m = 1$ and Θ_s is the threshold characterizing the novelty in the scene.

When the streakline distributions for all regions in the entire training sequence have been compared, the final dictionary of training activities can be defined as $\mathcal{D}_{train} = \{\mathcal{A}_1, ..., \mathcal{A}_n\}$ for these model assumptions. n is the number of activities identified from the training set.

The KL divergence is calculated on the test set similarly, with the exception that the models of the activities formed from the training set are no longer updated (but instances of the activity may still be found in the test set). A separate dictionary $\mathcal{D}_{test}$ is formed comprising of new activities found in the test set. $\mathcal{D}_{test}$ corresponds to new activities which are potentially anomalous formed only from the testing set. Note that the same threshold values θ and θ_s are used for both training and testing data.

4.6 Localisation of Activities

In the current approach, activity models are compared to other activity models identified in the scene during the training stage, regardless of their spatial location in the scene. This leads to an issue where if the scene contains strong perspective projection effects, then models close to the camera may appear significantly different statistically to those located further away, despite being a similar activity. Conversely, activities that are very different may appear similar given the perspective projection effects on the activity, for example, a cyclist towards the back of the scene may appear to have a similar statistical model to a pedestrian walking close to the camera. Furthermore, by comparing with all

activities in the scene, the current method does not consider that certain activities may be constrained spatially, for example, pedestrians may only walk on a path, and not across some grass area that is out of bounds. To alleviate these problems, we consider a method which only compares activity models within a certain spatial window. Such a spatial window must be dynamic to account for the perspective projection effect in the scene and its influence on the recorded human activity models. In this localisation approach, we define a dynamic distance model so that we can automatically adjust the size of the spatial window depending on the distance from the camera.

Firstly, we split the image frame into blocks of identical size as previously used for streakline estimation. For each block of pixels i we calculate the average of the movement magnitude intensity, calculated from all activities recorded in that area of the scene:

$$\Omega_{m,i} = \frac{\sum_j^{N_i} \mathcal{A}_{m,j}}{N_i} \tag{4.19}$$

where $\mathcal{A}_{m,j}$ represents the magnitude model for the activity taking place in the area corresponding to the block of pixels j and N_i represents all the activities identified in that sector of the image. We also consider the mean activity for the entire frame:

$$\Omega_m = \frac{\sum_j^{N} \mathcal{A}_{m,j}}{N} \tag{4.20}$$

where we consider all N activities identified during the training in the scene. The size of the region which is considered for the localisation of the movement is given by the ratio between the squares of the average movement magnitude in a certain area and the square of the average movement in the entire frame:

$$D(i) = b\frac{\Omega_{m,i}^2}{\Omega_m^2} \tag{4.21}$$

where b is a constant called base distance which weights the effect of local activity with respect to the total activity in the scene. This is particularly relevant in scenes where the perspective projection effect is strong and where the movement which is far away from the camera appears as smaller in intensity. Finally we only compare the activity k from the dictionary with a new activity if

$$|\mathbf{C}_j - \mathbf{C}_k| < D(i) \tag{4.22}$$

where $\mathbf{C}_j$ is the center point location of new activity j and $\mathbf{C}_k$ is the center point location of activity k from the dictionary. Consequently, the size considered for the area of localization

will be larger for regions characterized by larger movements. Regions which are further away, will be characterized by lower level of movement and consequently we would consider smaller regions of localization for such regions.

A distinct advantage of only comparing activities with others that satisfy equation (4.22) is that the computation time decreases significantly due to the significantly fewer comparison required using KL divergence. For example, given 100 activities in the scene and 5 activities inside the dynamic window, only 5 comparisons are required compared to the 100 required without the localisation approach.

4.7 Experimental Results

In this section, we evaluate the effectiveness of our proposed observational human activity identification methodology on state of the art datasets. The UCSD anomaly dataset [58] consists of two scenes observing university campus walkways. separate training and test sets, where it is assumed that normal human activity takes place in those video sequences used for the training stage while the video sequences used during testing are analysed for anomalous activities. The dataset consists of two scenes from a university campus: ped1 and ped2. Ped1 shows a university campus scene where pedestrians walk along an alley viewed under an oblique angle, the perspective distortion is quite strong, whilst in ped2 a wide alley is observed, in which pedestrians walk parallel to the camera plane and the perspective projection effects are minimal. Both scenes contain similar anomalous activities during the testing stage, such as: cyclists, electric vehicles, skateboarders and people walking in a different way (i.e. walking in a direction not observed in the training set, for example, across the grass). The anomalies are ground-truthed by the frame numbers of the anomalies and by indicating the spatial locations of the anomalies. The anomaly detection performance of the proposed method will be compared to that of the ground-truth. We also further evaluate the proposed approach on the UMN dataset. This dataset is simpler, consisting of staged group dispersing activities in an outdoor environment. Finally, we provide some observational experimental results on the i-LIDS Gatwick dataset[1]. The i-LIDS Gatwick dataset consists of cameras observing activity at Gatwick airport. Such scenes are very complex, and subject to many challenges such as pedestrian density, occlusions, perspective distortion and the inclusion of a wide variety of complex

[1] https://www.gov.uk/guidance/imagery-library-for-intelligent-detectionsystems

yet subtle activities.

Streaklines and Segmentation Evaluation

To begin, the single/multi-vector streaklines methodology is applied to the video sequences. The first consideration is the size of the spatio-temporal macro-block for the streakline methodology. Note that the same size blocks will be used for both the block matching approach and the streakline methods, to enable a fair performance evaluation. Spatio-temporal blocks which are too large will not capture the essential motion representing the human activity, whilst blocks too small will incur too much noise and capture very small movements that may be unnecessary and add unnecessary complexity to the model. The macro block size M from equation (4.1), was set empirically after some testing across the different datasets. A macro block size of $M = 10$ pixels was deemed appropriate for the UCSD and UMN datasets, considering the size of the pedestrians, cyclists and electric vehicles which are observed in the analysed scenes. To increase the amount of motion statistics, overlapping blocks are used. An overlap of half is used, providing 4 times the amount of statistics, essentially providing a streakline model for each 5×5 block of pixels. For the i-LIDS Gatwick dataset, we consider blocks of 20×20 pixels, considering the higher video resolution of this dataset. Since the streakline vectors are extracted over spatio-temporal blocks, the temporal window (streakline length) must be considered. In this case, we set the length of the streaklines T to 10 frames for all datasets. This provides reasonable medium-term flow coverage without causing the streakline trajectory to degrade. A further consideration at this stage is the removal of noisy/erroneous motion estimation vectors. To alleviate this, we use a simple threshold filter, removing any vectors below an intensity streakline magnitude of 1, which is sufficient to remove background noise and erroneous vectors. Later, when segmenting the moving regions, we also consider removing any very small region, which could actually correspond to noise.

An example of the motion estimation using the block matching is shown in Figure 4.2a. In this example, the motion of the pedestrians walking in different directions is well captured. The faster motion of the cyclist is well captured by the block matching and the motion vectors are visibly larger (indicating quicker movement). In the example in Figure 4.2a there are a few erroneous vectors. Such erroneous vectors are due to the presence of MPEG compression artefacts in the video and because of large smooth image areas. These erroneous vectors will be removed by the threshold filter discussed above.

After the motion estimation is complete, the connected components are segmented and labelled. As described in Section 4.4, this is done by a simple algorithm which passes through the image, segmenting the video into regions of specific movement. This step is essentially an application of the well-known grass-fire algorithm in which objects are segmented recursively, by checking the movement vector connectivity and their similarity in the neighbourhood. At this stage, each labelled region may contain more than one distinct behaviour, for example a cyclist and a pedestrian moving in opposing directions. After the initial segmentation is completed, regions that are considered too small are removed. In this case, regions smaller than 5 blocks in size are removed. This size was chosen as regions smaller than 5 blocks do not represent any meaningful region of motion. The motion that is represented by less than 5 blocks is almost always either noise (from erroneous vectors) or is not relevant (e.g. small movement of the feet or a bag).

In the example in Figure 4.2c separate regions of motion are detected. In region 1 in Figure 4.2c, all pedestrian are walking in the same direction, therefore it is expected that only one activity is detected in this region. In regions 3 and 5, there is only a single activity as only one pedestrian is present in each region. In region 7, there are at least 3 different activities - the cyclist moving left, 2 pedestrians moving right, and a single pedestrian moving left. Region 9 is slightly more complex, at least 2 separate activities are expected, perhaps even 3. The pedestrian walking to the right will form one activity, while the pedestrians walking to the left should form a second activity. However, given that one of the pedestrians walking to the left in region 9 is walking at an oblique angle, this may form a 3rd activity, depending on the individuals motion. Ideally, 3 different activities should be detected in region 9.

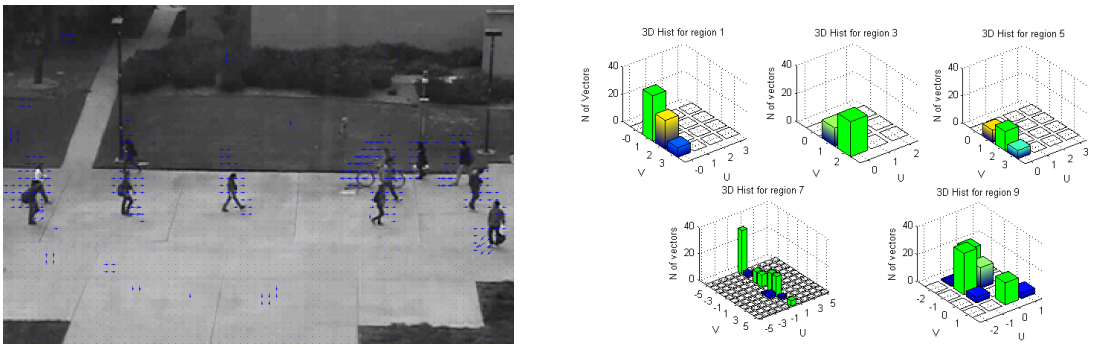
Histograms of the movement vectors are generated for each movement region in the image, obtained from the single streakline/block matching-based methodology of each region. Peaks are detected by way of detecting maximas in the histograms, i.e peaks that are greater than their neighbours. The EM Algorithm is applied (under the Gaussian modelling assumption) to each region of motion, defined by their inter-connected regions. In the case of multiple histogram peaks (multiple modes), the number of peaks in the histogram is used for defining the number of Gaussian mixture components when initialising the EM algorithm. As described in Section 4.4, we introduce the location parameters of the region as an extra component into the EM algorithm to help improve the segmentation of the regions. By introducing such location parameters, emphasis will be placed on

clustering blocks that are nearby in space, avoiding 'broken' regions and resulting in well defined movement region boundaries.

As described in Section 4.4, we adopt two different approaches of movement segmentation for defining activities. The first approach consists of global segmentation, where each GMM component corresponds to a certain movement. This means that several regions of movement from the frame may correspond to the same activity. For example, a pedestrian walking to the right in one part of the scene may correspond to the same GMM component as a pedestrian walking to the right in the other side of the scene. In the second approach, we adopt a local approach, where each region of movement is defined locally by considering its location in the frame. In this case, each interconnected region is considered as a multivariate GMM and each GMM component corresponds to the segmented regions inside that interconnected region. Following on from the example above, the two pedestrian would correspond to two different GMMs. While the global approach will produce fewer but spurious regions of activity, the second approach will produce additional regions of movement, each of them compactly defined in the space of the video frame.

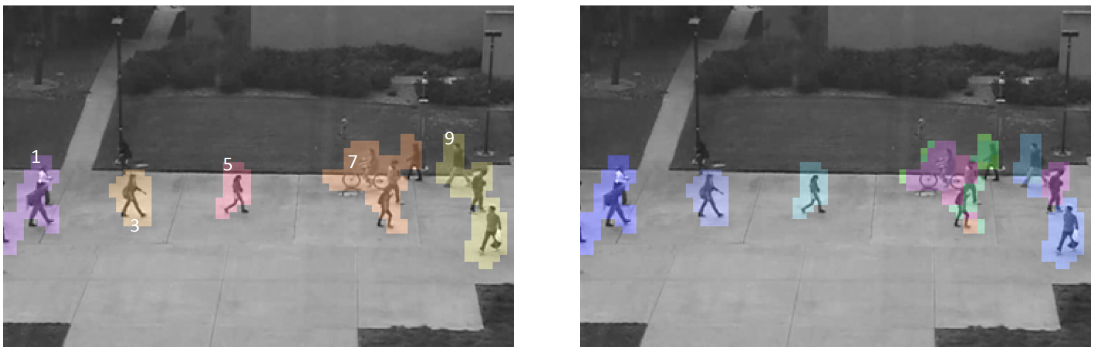
The expected (ideal) results for the histograms (from the segmented regions example in Figure 4.2c): Region 1,3,5 with 1 peak each and Region 7 and 9 with 3 peaks each. From the histograms in Figure 4.2b, it is clear that regions 1, 3 and 5 all have just a single peak each as expected. In the case of region 7, 5 peaks are detected. It can be observed that the cyclist is still visibly separated from the pedestrians. The cyclist has clearly moved notably faster than the pedestrians. In the example of the segmentation in Figure 4.2d, regions 1, 3 and 5 are no different from the original segmentation (single distinct regions each). Region 7 is more complex - the cyclist is well represented and so is the pedestrian moving to the right. The moving regions of the other two pedestrians are not represented that well as distinct moving regions. This is not necessarily a problem as the main goal is detecting the abnormal activity, which in this case would be the bicyclist. The smaller regions on the edges of region 7 will also be removed at the next step. Region 9 is well represented and 3 distinct regions are identified for each of the activities. Following the segmentation, small regions (under 5 blocks in size) are removed in the same manner as prior to segmentation. In this case, the regions around the bicyclist and the pedestrians (in region 7) will be removed. This is beneficial as such regions do not represent any meaningful parts of a moving object.

Given such segmentation, each moving region is represented by its multi-variate Gaus-



a) Motion estimation

b) Histograms of motion flow



c) Inter-connected regions (pre-segmentation)

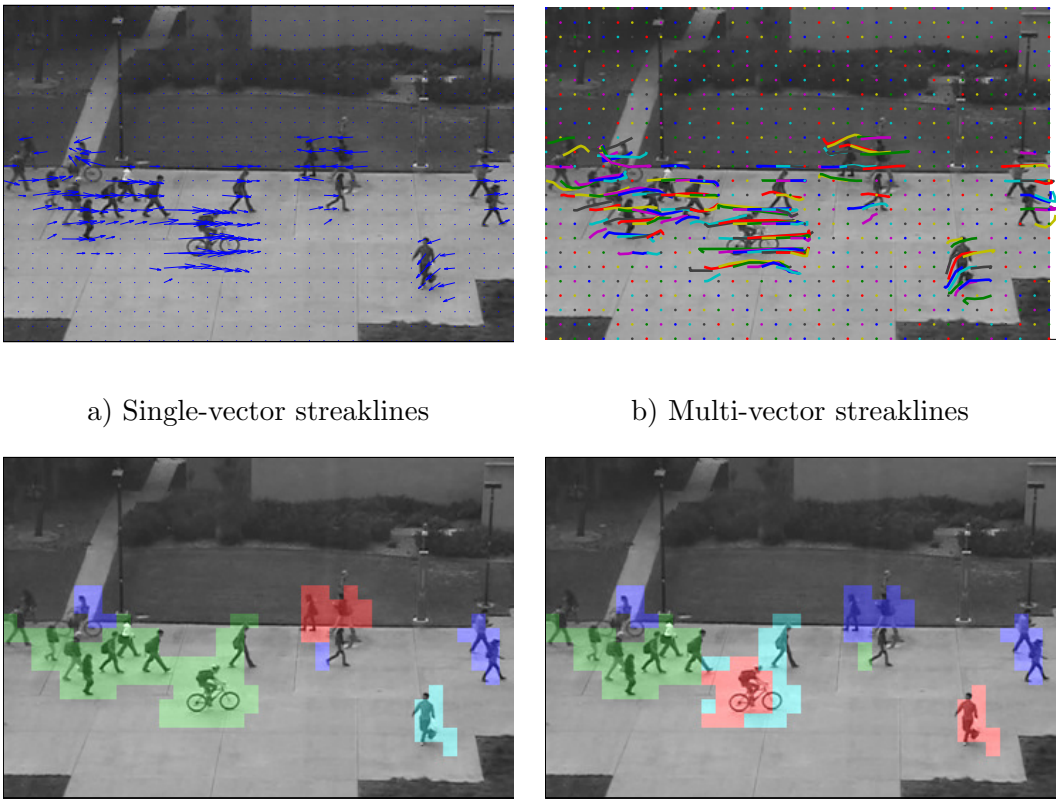
d) Moving regions (post-segmentation)

Figure 4.2: Example of the application of motion estimation, the corresponding motion histograms and the moving region segmentation. Example sequence from ped2 of the UCSD dataset.

sian mixture model representation of the single or multi-vector streaklines, as described in Section 4.3. Figure 4.3 shows an example of single-vector streaklines, multi-vector streaklines and their corresponding segmentation results achieved on ped2 of the UCSD dataset. It can be observed from Figure 4.3a and b, that both single and multi-vector streaklines provide a good representation of the motion flow. The multi-vector streaklines in Figure 4.3b provides a more detailed representation of the individuals' motion when compared to the single vector streaklines in Figure 4.3a. Furthermore, the movement of the individual in the right portion of the scene is better captured by the multi-vector streaklines. It can be observed from Figure 4.3c and d, that multi-vector streaklines provides better identification of the cyclist from centre left of the scene, when compared to the result achieved with single vector streaklines. This suggests that the multi-vector streaklines may perform better at localising activities in the scene, but may capture irrelevant movements that could degrade detection performance.

A further example of the streaklines and segmentation is shown in Figure 4.4. This example is from ped1 of the UCSD dataset, where pedestrians are observed over a pathway at an oblique angle showing strong perspective projection effects. In this example, several anomalous activitivities, such as those corresponding to a cyclist, a peson pushng a cart and a skater, can be observed. However, the cyclist in the bottom left of the screen is not segmented separately in the single-vector segmentation in Figure 4.4c, while in Figure 4.4c, the same cyclist is well segmented. This again demonstrates the potential advantage of multi-vector streaklines for localising such activities. In Figure 4.5, the groundtruth is shown for the anomalous activities corresponding to the same scene as shown in Figure 4.3 and Figure 4.4. The segmented regions shown in Figure 4.4d match those of the ground truth (in Figure 4.5b) for both the cyclist and the person pushing a cart. The anomalous moving region corresponding to the person from further end of the path, located in the upper-right corner of the scene is merged with the moving region of a nearby pedestrian due to the limited view and perspective distortion present in the scene. Merging such regions may not be a problem if such a region is identified as anomalous regardless. Furthermore, the segmentation previously presented in Figure 4.3d match closely to those of the ground truth provided in Figure 4.5a. This is promising for the activity localisation results on such datasets.

Given the streakline GMM models for each region, the next stage is comparing the activities statistically using the KL divergence. During this stage, the parameters cor-



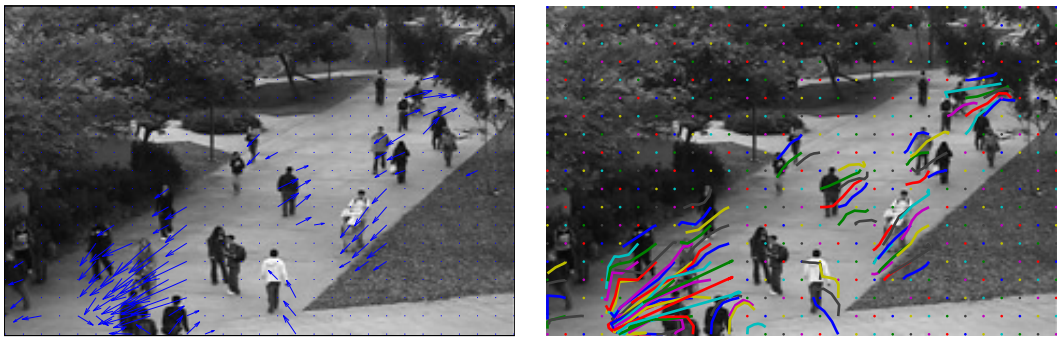
a) Single-vector streaklines

b) Multi-vector streaklines

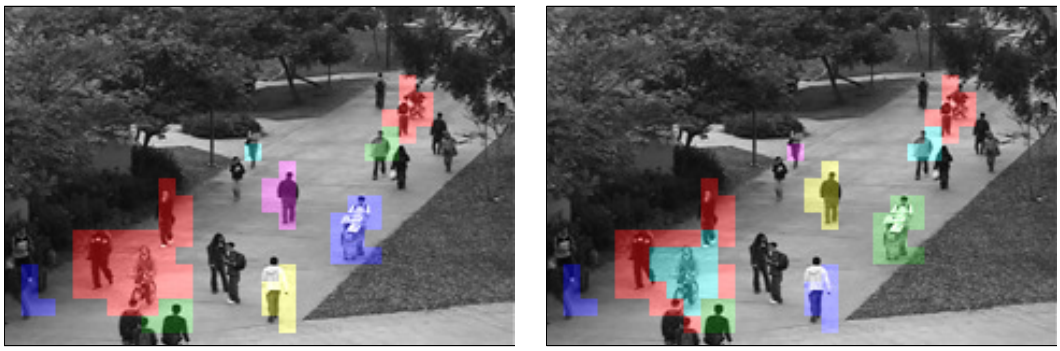
c) Single-vector segmentation

d) Multi-vector segmentation

Figure 4.3: Modelling movement using streaklines on a video sequence form the ped2 UCSD dataset.

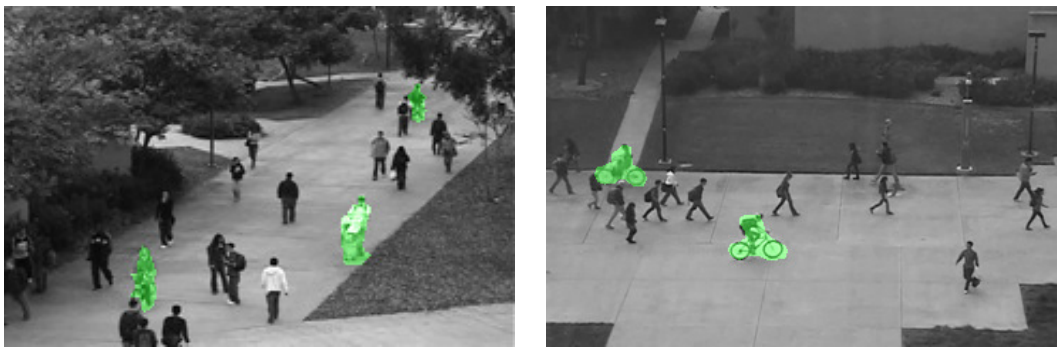


a) Single-vector streaklines b) Multi-vector streaklines



c) Single-vector segmentation d) Multi-vector segmentation

Figure 4.4: Modelling movement using streaklines on a video sequence form the ped1 UCSD dataset.



a) Ped1 UCSD dataset b) Ped2 UCSD dataset

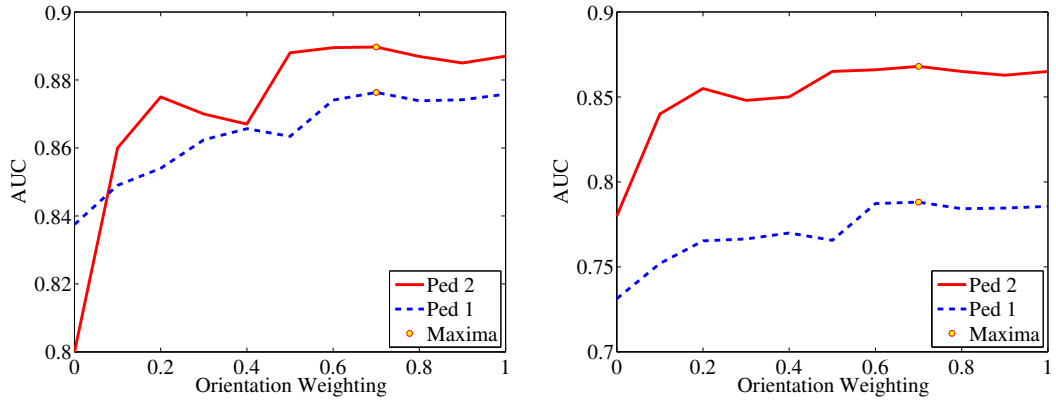
Figure 4.5: Example of the groundtruth for anomalous activities from the ped1 and ped2 UCSD datasets.

responding to the observed activities are compared to those representing the activities recorded in the dictionary of activities during the training stage. In the following we evaluate the results provided when using the polar coordinates by varying the threshold Θ_s from equation 4.18 when deciding new activities. We compare the area under the ROC curve (AUC) when varying the orientation component $k_o \in [0, 1]$ in equation (4.18), whilst the magnitude component is $k_m = 1 - k_o$. Figure 4.6a and Figure 4.6b, show the results in activity detection and localisation, respectively, when increasing the weight of the vector orientation and decreasing the weight of the movement amplitude, when using multi-vector streaklines. For the detection performance, we assess the percentage of frames where the anomalous activity is found, against the labelled ground-truth. For assessing the localisation, the detected region of activity is compared to that of the corresponding groundtruth mask (measured as a a percentage of detected pixels out of the groundtruth labelled pixels). It is clear from Figure 4.6 that the best results are obtained when the orientation weighting is $k_o = 0.7$. This also suggests that a small improvement in the results is expected when using polar coordinates over Cartesian coordinates as the best results are obtained when the weighting between orientation and magnitude are not equal.

Next, we evaluate the effect of using localization when comparing activities only within a certain neighbourhood as described in Section 4.6. To evaluate this, we consider the difference in localisation result performance as the base distance b, from equation (4.21), is varied in the range $b \in [40, 150]$. Figure 4.7 shows the localisation performance rate as the base distance b is varied. Notably, there is a clear improvement when using a smaller base distance over using the maximum base distance ($b = 160+$, no localisation). The best results are obtained when the base distance is $b = 90$. This provides a 6% improvement in localisation results when compared with using no localization (when $b = 160+$).

4.7.1 Anomaly Detection Evaluation

The anomalous activity detection by the proposed method is evaluated in three ways: by monitoring visually the activities detected in the scene, by means of timing correctly the activities by identifying the frames where they occur and by identifying the location of the activity in the scene. For the UCSD dataset, ground-truth is provided by way of frame numbers for the anomaly detection in the test set, and by a spatial ground-truth mask indicating the location of the anomalous activities. The anomalous detection result provided by our method are evaluated against these groundtruths to provide a numerical



a) Detection performance b) Localization performance

Figure 4.6: Evaluation of the AUC (area under the ROC curve) when varying the orientation weighting k_o for the UCSD dataset.

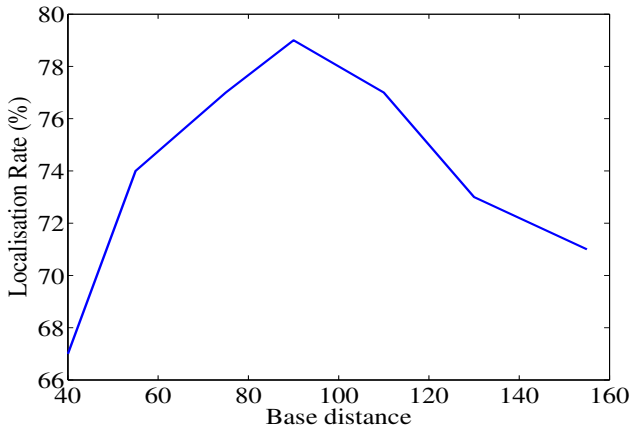


Figure 4.7: Localisation performance when the base distance b is varied, in Ped1 and Ped2 from UCSD dataset.

analysis of the results.

To begin, we introduce a visualisation of the activities over time called the 'Activity Monitor'. Activity monitors are used to show the activities identified in the scene according to their timing of occurrence in the video sequence. The first example of an activity monitor is shown in Figure 4.8. For reference, the yellow colour are frames with activities present and the red coloured frames indicate no activity present. For the bottom row (groundtruth), the red ground-truth colour is used when no anomalous activity are present in the scene and the yellow colour is used when anomalies are present in the scene. The groundtruth indicates only certain anomalous activities, such as the presence of cyclists or skaters, which should not be allowed in the scene. For reference, example frames from test sequence 2 are shown in Figure 4.9. We can observe the appearance of a cyclist in the scene in Figure 4.9b and the inclusion of the cyclist in Figure 4.9c.

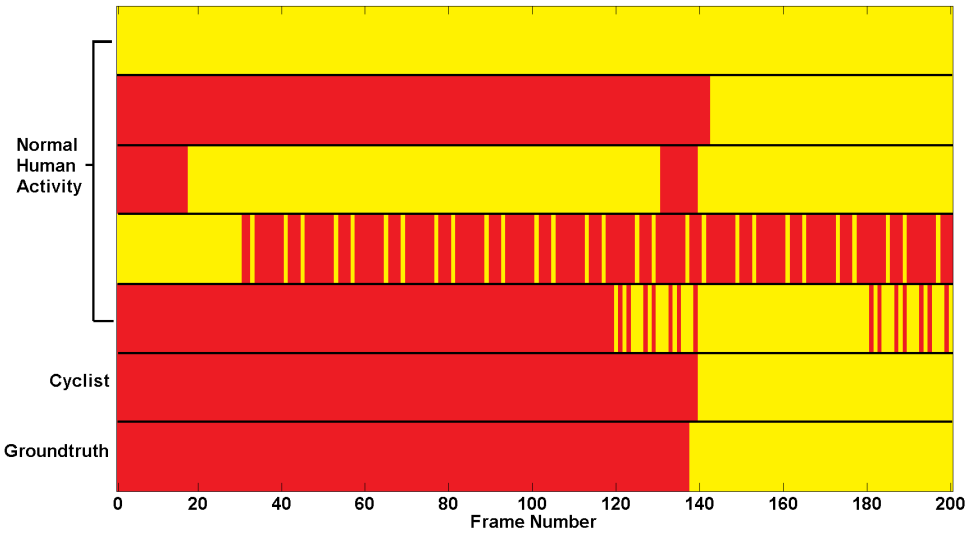


Figure 4.8: Activity monitor for test sequence 2 of the UCSD ped 2 dataset.

a) Frame 1 b) Frame 95 c) Frame 149

Figure 4.9: Example frames from test sequence 2 of the UCSD ped 2 dataset.

We can observe from Figure 4.8 that there is a single anomalous activity representing the cyclist who enters the scene at around frame 93. The activity is correctly detected by this method for the whole duration of the scene. Note that the cyclist is detected before the ground-truth declares the anomaly; this is because a small portion of the wheel is present and this triggers immediately the presence of an anomalous activity by our method, where as the ground-truth is set such that a certain amount of the anomalous region must be present in the scene before it is declared as anomalous. This indicates that the method performs very well, even when the anomalies are small such as only a portion of bicycle wheel being present. Our method could be changed such that the region must be of a certain size before anomalies are detected; this would bring the detection results in-line with the ground-truth. The normal activities in Figure 4.8, activities (1-6), are representative of different types of walking activity - walking left, walking right and so on. Activities 2-4 represent most of the walking in the scene. The walking activity is not continuous, with frames where the activity is not present, as it can be observed for the activities 2-4 from the activity monitor in Figure 4.8. This is not necessarily a problem, but a more constant walking detection without the stagnation would be preferred. This could be achieved by simply lowering the threshold for detecting new activities; but this would change the sensitivity for detecting anomalies, therefore must be considered carefully.

In the second example, shown in Figure 4.11, two anomalies are present in the scene. Once again, examples frames from the sequences are shown in Figure 4.10. In this example, anomalous activities are clearly present in all 3 frames. For colour reference, the activity monitor in Figure 4.11 is coloured as follows: the light blue colour are frames with activities present and the darker blue colour are frames with no activity present. The red ground-truth colour is used when anomalous activity is present in the ground-truth and the cyan colour is used when anomalies are present in the scene. In this example, the cumulative anomaly is also shown so that the result can be directly compared with the ground truth. The cumulative anomaly represents the time where any anomalous activity is detected by the method. The cumulative anomaly is shown to allow easy comparison between the anomaly detection and the ground-truth. A bicyclist is already in the scene at the beginning and leaves the scene at around frame 50-60. A second cyclist enters the scene at around frame 19, and remains in the scene for the entire duration of the sequence. This means that the ground-truth will always indicate an anomalous activity. In Figure 4.11, the cyclists are detected entering and leaving the scene correctly. In this more complex

case, the first and second cyclist aren't always detected in the scene - this is because at certain points the cyclists are severely occluded by pedestrians. Interestingly, a third anomalous activity is detected. This activity is a variation of the first cyclists activity and as the cyclists activity changes throughout the scene, the detection switches between the initial cyclists activity and the other anomalous activity (both are detected in the activity monitor, as it can be seen from Figure 4.11). Note that although the anomalous detection of each cyclist is not perfect in this example, the cumulative anomaly is still correct as per the ground-truth.

a) Frame 19 b) Frame 36 c) Frame 90

Figure 4.10: Example frames from test sequence 9 of the UCSD ped 2 dataset.

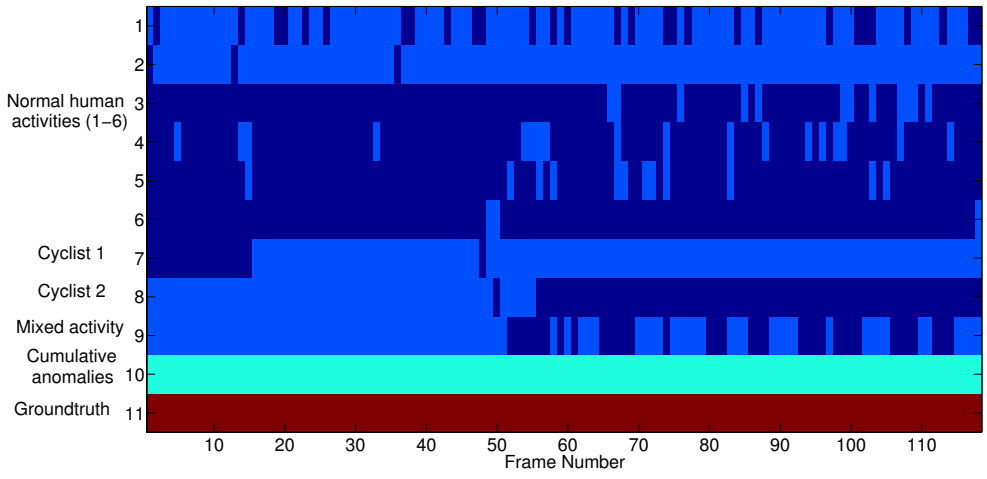


Figure 4.11: Activity monitor for test sequence 9 of the UCSD ped 2 dataset.

Numerical Evaluation

In the following we provide the numerical evaluation results obtained when using the proposed observational human activity identification methodology on three data sets: the UCSD dataset, the UMN dataset and the i-LIDS Gatwick dataset. As previously men-

tioned, the UCSD data set consists of two scenes from a university campus: Ped1 - where the pedestrians walk along an alley viewed under an oblique angle and Ped2 - showing a similar environment, but where the pedestrians walk parallel to the camera plane. Due to the viewing angle of Ped1, the video representation of the scene under view is affected quite strongly by perspective distortion. Ped2 also contains some perspective distortion, but due to the pedestrians walking in parallel to the camera plane, it does not affect the results to any measurable degree. The UMN dataset is simpler than the UCSD dataset, consisting of staged abnormal activity where individuals behave normally followed by a sudden dispersing/panic activity from the pedestrians in the scene. Three different dispersing activity scenes under different lighting conditions and environments with varying number of individuals from the UMN dataset, are considered for the experiments. The i-LIDS Gatwick dataset, as described earlier, consists of video sequences from cameras observing Gatwick airport. The cameras observe scenes which are complex in nature; under varying light conditions together with perspective distortion. The density of the pedestrians in the scene also vary, and many of the activities shown in these videos are complex.

We begin by discussing the numerical results of the proposed methodology on the UCSD dataset. To evaluate the proposed method, we follow the evaluation scheme defined by [58], where the performance of the method is examined by its detection and localisation performance on both ped1 and ped2 of the UCSD dataset. The detection performance is evaluated by determining if a given frame in the test set is correctly labelled as anomalous activity according to the corresponding ground-truth. The localisation performance is evaluated by determining if the segmented regions of the anomalous regions matches those indicated by the ground truth segmentation masks. A correct localization result is achieved when at least 40% of the pixels are correctly identified as part of a region defined as corresponding to uncharacteristic movement in the scene [58]. We evaluate the method by its true positive rate and false positive rate, both for detection and localisation results. To compute such results, we evaluate our method on the whole UCSD dataset as the threshold Θ_s is varied. The ROC (receiver operating characteristic) curve displaying the true positive rate against the false positive rate for the entire datasets, when the threshold of the KL divergence from equation (4.18) is varied in the range $\Theta_s = [0, 5000]$. The ROC curves for the results on the UCSD dataset are shown in Figures 4.12a and b for temporal and localisation detection in Ped1, whilst the corresponding results for Ped2

are shown in Figures 4.13c and d respectively. The 'local/global optical flow' methods refers to the block matching based optical flow method with the local or global segmentation, respectively, described earlier in Section 4.3. The 'single vector streaklines' method refers to the single-vector streakline methodology with local segmentation, while the 'multi vector streakline' method refers to the multi-vector streakline methodology, with local segmentation, polar coordinates and the localisation method described in Section 4.3. The numbers in brackets refer to the area under the ROC curve (AUC), providing a measure of performance across the range of sensitivities when the threshold is varied. Overall, the streakline based methodologies outperform the local and global optical flow methods. Interestingly, the multi-vector streaklines outperform the other methods at localisation, while the single-vector streaklines perform better at detection results. This is expected as earlier in this chapter it was noted that the segmentation of anomalies were notably better when using multi-vector streaklines. It is suggested that the single-vector streaklines perform better at detection tasks due to the added complexity of the multi-vector streaklines causing some confusion when individuals perform small complex movements which despite being well captured by multi-vector streaklines (and lost in the single-vector model), do not add any additional useful characteristic data to the abnormal activity model. Notably, the overall results on ped1 are worse than those of ped2; this is expected due to the perspective distortion affecting ped1 and the limited view of the anomalous activities when they take place towards the end of the path.

As previously discussed, we also evaluate the performance of the KL divergence in this context by comparing the results of the methodology using KL divergence to the Jensen-Shannon divergence and the traditional Euclidean distance. In this case, we use the single vector streaklines on ped2 dataset and report the best obtained detection recognition rate for each statistical measure in Table 4.1. The best performing statistical measure in this case is clearly the symmetric KL divergence. Considering this, we continue to use the symmetric KL divergence for the final experiments in this work.

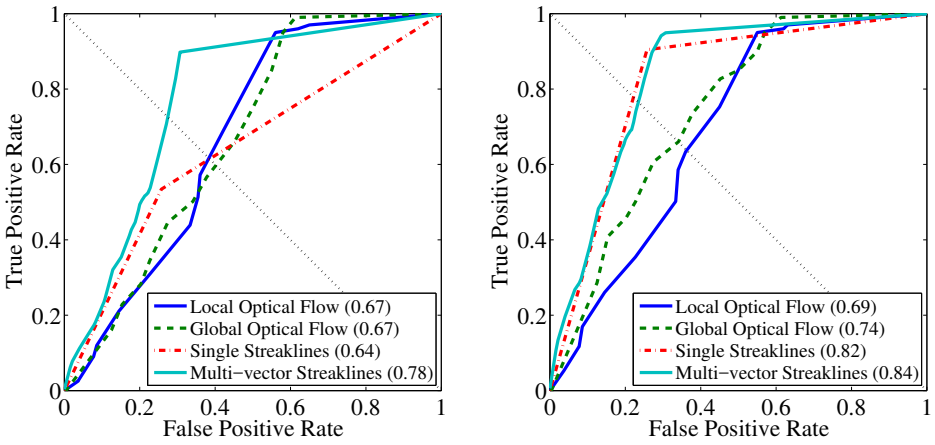
Table 4.2 provides the results for the detection and localisation evaluation of the proposed methods when compared with the state of the art. The best detection of uncharacteristic movements is obtained for the threshold where the minimum equal error rate (EER) is lowest. The detection and localisation columns indicates the best performance of the method (1-EER). Almost all the streakline results show a clear improvement in results over the global/local optical flow. It is again clear from the table that the multi-vector

streaklines generally perform better at localizing human activities, whilst the single-vector streaklines are better at detecting the frames when the anomalous activities take place. Notably, the polar coordinates and localisation show an improvement in results of the multi-vector streaklines on both ped1 and ped2. A greater improvement is noted on ped1 (+6.6%) compared to ped2 (+0.5%) when localisation and polar coordinates are added; this is expected due to the localisation methodology having a greater effect on ped1 where the perspective distortion is far more prominent. The multi-vector streaklines perform better at the localisation task than the state of the art methods on both ped1 (+10% improvement) and ped2 (+0.8% improvement). The detection results are on par with state of the art, although the methods that perform better at detection do not provide their localisation results; suggesting that their localisation results are not on par with their detection results.

Next, we evaluate the proposed methodologies on the UMN dataset [66]. This data set consists of 11 video sequences of three different scenes. Each sequence follows the same staged format, where the sequence begins with normal behaviour consisting of individuals walking around, followed by an abnormal section where individuals run/panic in the scene. The anomalies are therefore considered global events in the scene and do not require any form of individual localisation. The sequences are on average around 20 seconds long, where the normal section is usually longer than the abnormal section. The video sequences have a resolution of 320 by 240 pixels, shot in a variety of scenes including outdoors with good lighting and indoors with poor lighting. In these sequences, we learn the normal activity by randomly selecting 20 frames from the normal portion of the sequences. Finally, we test these models on the remainder of the frames. Examples of three frames where anomalous behaviour is detected are shown in Figure 4.14. Note that all three scenes are in different environments, under a different camera perspective with different lighting. In Table 4.3, we provide the numerical performance results of our method by computing the area under the curve (AUC). All state of the art methods currently perform extremely well on the dataset; this is expected as the activities are simple and staged. Our results for both single-vector streaklines and multi-vector streaklines are on par with those from state of the art in Table 4.3, and in some cases show an improvement. Due to the slight differences in evaluation protocols on this dataset, some variability in the results are expected, however all of the detection performance results are already extremely high, limiting any potential performance improvements.

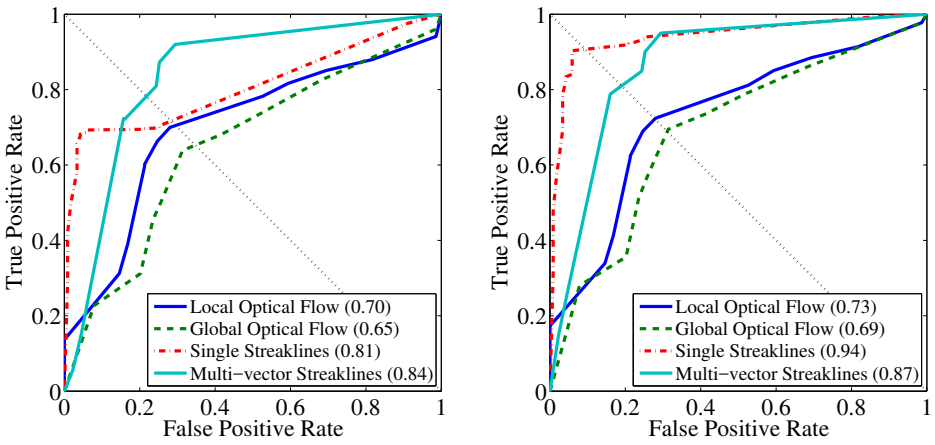
Finally, we provide some basic anomaly detection results on The i-LIDS Gatwick dataset[2]. As mentioned earlier, the i-LIDS Gatwick dataset consists of video sequences obtained from CCTV cameras placed around Gatwick airport. The resolution of the video sequences is 720×576 pixels. The video sequences contain many complex activities, with mixed crowd densities, together with many other difficult challenges such as floor reflectance under changing lighting conditions and multiple occlusions. Given the lack of groundtruth for abnormal activities on the dataset, we instead provide results from a scene where 'normal activity' consists of walking out of the exit terminal and "abnormal activity' consists of walking against the flow. Our training data set contains 30 seconds of video from the normal exit behaviour corresponding to persons leaving the terminal. Our testing data set consists of 10 seconds of video containing a mix of normal behaviour and pedestrians attempting to walk against the flow. The movement flow and the activity segmentation, calculated from the streaklines is provided in Figures 4.15a and b for the training stage and in Figures 4.16a and b for the test stage. The convergence of the streakline directions in the center-left part of the images from Figure 4.15a and Figure 4.16a is produced by the perspective projection effect when streaklines are used. Their converging direction indicates the direction of movement for the persons under observation. The different activities are coloured differently in Figure 4.15b and Figure 4.16b. Two persons carrying luggage trolley (coloured red), which are not present in the training set are detected as a new activity in the test set, as it can be observed in the segmentation of activities in the scene from Figure 4.16b. However, the individual pulling a suitcase to the right of Figure 4.16b is detected as blue (normal walk left). Notably, other activity classes are reasonably well separated considering the limited training data and complexity; for example, pedestrians walking to the left without luggage group together to form one class (visualised in blue), while individuals walking to the right form another class (visualised in green). Furthermore, a single block forms another activity as seen in the right of the image in Figure 4.16b. This single block would subsequently be removed when eliminating very small regions corresponding to noisy optical flow or to movement corresponding to movements which are not deemed as significant.

[2]https://www.gov.uk/guidance/imagery-library-for-intelligent-detection-systems



(a) Localization ped1 (b) Detection ped1

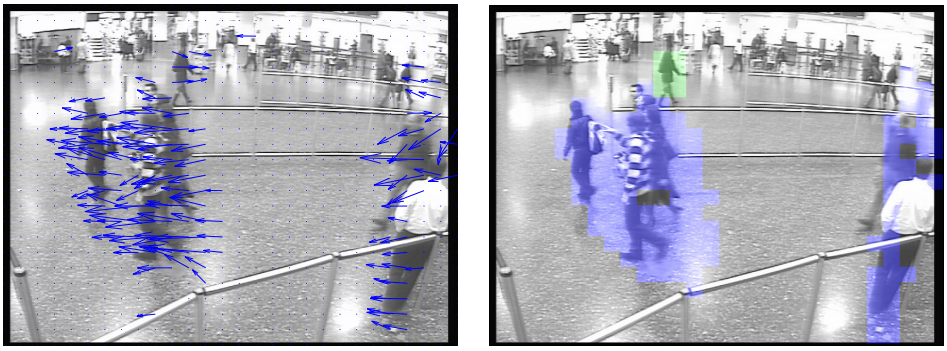
Figure 4.12: ROC curves when varying the threshold Θ_s for local flow, global flow, single and multi streaklines methods when applied to ped1 of the UCSD dataset.



c) Localization ped2 d) Detection ped2

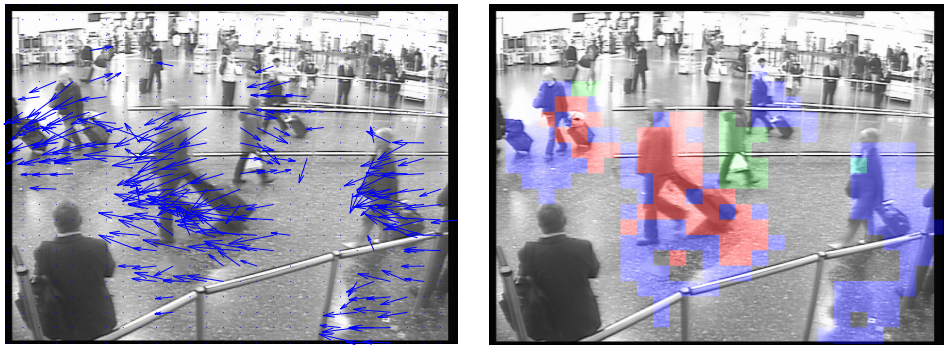
Figure 4.13: ROC curves when varying the threshold Θ_s for local flow, global flow, single and multi streaklines methods when applied to ped2 of the UCSD dataset.

Figure 4.14: Frames from the UMN dataset, where anomalous behaviour has been identified.



a) Training flow b) Training segmentation

Figure 4.15: Example of training data from the i-LIDS Gatwick dataset. Sequence from the exit terminal shown.



a) Testing flow b) Testing segmentation

Figure 4.16: Example of testing data from the i-LIDS Gatwick dataset. Sequence from the exit terminal shown.

Table 4.1: Detection results using different statistical measures on ped2 of the UCSD dataset using single vector streaklines.

Statistical Measure	Detection Rate (%)
Symmetric KL divergence	90.4
Jensen-Shannon divergence	81.1
Euclidean distance	56.3

Table 4.2: Numerical anomaly results for the UCSD dataset.

Method	Ped 1		Ped 2	
	Detection (%)	Localisation (%)	Detection (%)	Localisation (%)
SF [66]	69.0	21.0	58.0	-
MPPCA [50]	60.0	18.0	70.0	-
MDT [58]	82.2	64.8	81.5	70.1
LE [105]	78.0	69.0	86.5	78.0
GMM-MRF [72]	85.1	-	95.1	-
Global Optical Flow	66.2	59.2	69.4	63.9
Local Optical Flow	64.7	62.4	72.3	70.2
Single Streaklines	77.9	62.2	90.4	70.3
Multi-vector Streaklines	76.6	72.5	81.0	78.3
Mutli-vector Streaklines with polar coordinates and localisation	79.8	79.1	81.4	78.8

Table 4.3: Comparison in abnormal activity recognition results using the area under ROC curve on the UMN dataset.

Method	Scene 1 (%)	Scene 2 (%)	Scene 3 (%)
Cong *et al.* [24]	99.5	97.5	96.4
Shi *et al.* [99]	93.6	77.5	96.6
Thida *et al.* [105]	98.0	98.0	97.0
Single Streaklines	99.1	96.5	97.3
Multi-vector Streaklines	95.3	89.7	96.6

4.8 Conclusions

In this chapter, we presented a new approach to observational abnormal human activity identification from real world video sequences. In this approach, both syntactical and statistical modelling of short-to-medium level tracking is employed by using streaklines to represent the movement flows of individuals in the scene. Human movement was segmented using the EM algorithm under the Gaussian Mixture Model assumptions. Segmented moving regions were represented both in movement and location space by their streakflow and location models. Two different approaches to streaklines are presented: single-vector streaklines and multi-vector streaklines. In the single vector approach, PCA is utilised to project the principal movement vector representing the movement over several frames. In the multi-vector streakline approach, the magnitudes and directions of the streakline are characterised by a single magnitude vector spanning multi frames and multiple direction vectors. The magnitude and direction vectors are represented using polar coordinates, where a weighting factor is introduced to balance the magnitude and direction. Furthermore, a localisation methodology is introduced in order to account for the perspective projection present in the scenes. In this approach, activities are only compared with other activities within a given dynamic window, computed based on the motion vectors and distance from the camera. A dictionary of activities is then generated for the training data, recording statistics of both intensity and direction of movement as well as the location coordinates of the moving region, characterizing the scene. During the testing stage, the symmetric KL divergence is used to compare statistically the observed movement with those recorded in the dictionary. While the single-vector streakline approaches provided good results at detecting the new human activities, the multi-vector streakline approaches performed better at spatially localizing such activities in the scene.

One issue with the current work is that complex activities in the scene are often misunderstood or not captured at all. Furthermore, the interactions within pairs or groups of people are not modelled and such a task is incredibly important for detecting abnormal activities in complex scenes, where group activities are more prevalent than individual atomic activities. Next, we propose to improve our methodology to detect complex group activities in real-world environments.

Chapter 5

Group Activity Recognition

5.1 Introduction

In this chapter we analyse the activity of groups of people and how individuals interact with each other. Group activity recognition has attracted sufficient interest only recently, despite being essential in defining the real intention and overall scene context of human activities. The area of group activity recognition has a significant importance especially for video surveillance among many other applications including semantic annotation of videos and automatic video retrieval.

In comparison to human activity recognition, group activity recognition requires more complex descriptions of the group as a whole in the context of the given scene. Some methods assessing the activity of groups of people from video sequences have been recently proposed, for example, Ni *et al.* [75] recognised group activities using localized causalities based on manually initialized tracklets. Lin *et al.* [60] used a heat-map based algorithm for modelling human trajectories when recognising group activities in videos. Chang *et al.* [18] used a probabilistic approach to group human activity by modelling the movement tracks between interacting individuals using a multi-camera system. Choi *et al.* [23] proposed a framework for analysing collective group activities based on different levels of semantic granularity. Zhang *et al.* [125] addressed the problem of group event recognition by computing histograms of different features extracted from tracklets, representing localized movement in the video. Similarly, Cheng *et al.* [20] modelled group activity as a framework composed of multiple layers and Gaussian processes were used for representing motion trajectories. These methods rely on either the training of a pedestrian detector for each scene, or some manual initialization of tracklets. This is impractical in

the real world, especially when pan-tilt-zoom (PTZ) cameras are used where the camera and scene parameters may change. For example, panning and zooming would change the perspective projection and the scene visible through the camera. Another issue with the proposed methods is that each pedestrian in the scene is often treated as a single entity represented by a tracklet, as opposed to considering that a single pedestrian may be composed of several moving atomic events, as for example would be the case when an individual is performing a more complex activity such as fighting. While some methods such as in [20] aim to address this issue by using appearance feature. Such appearance features are often too specific to the individuals and often over-fit the model to the individuals specific attributes such as body shape and clothing rather than modelling the actual human activity recorded in the video sequence.

This chapter proposes an automatic method for group activity recognition by modelling the inter-dependent relationship between features over time. Unlike in the other methods described above, the proposed method does not rely on any manual initialisation of tracklets and instead makes use of automatically extracted streakflows to represent the movement of regions over several frames. The interdependency between moving regions is represented by evaluating the relative movement and location of each moving region with respect to all the others in the scene at a particular time instance. The dynamic changes of the inter-dependency of the features are also modelled by considering the differences between features over certain intervals of time. The change in interdependency between moving regions is modelled over time by using Kernel Density Estimation (KDE) to model the change of movement and location in time, for various participants to the movement identified in the video sequence. The model also keeps track of the locations of stationary pedestrians by marking the locations where they stop moving in the scene. The proposed method also introduces a scaling procedure to compensate for the effect of perspective projection in video sequences which is evident in the case of video recordings by cameras of wide angle located at low heights, which is a very common occurrence in video surveillance data.

The remainder of this chapter is organised as follows: Section 5.2 describes the modelling of streakflows and location features used for representing moving regions. Section 5.3 describes the modelling of the inter-dependencies between the motion and location features. Section 5.4 describes how the inter-dependencies between features are represented over time using KDE and describes the classification of group activities. Section 5.5

presents the comparative results for the proposed method and finally Section 5.6 draws the conclusions of this research work.

5.2 Group Activity Modelling

The proposed methodology has several stages which are shown in the block diagram in Figure 5.1. To begin, the streakflows are extracted to represent medium-term flows of movement. Following this, the streakflow is segmented using a scaling factor derived from the initial segmentation estimation. After identifying and modelling the movement of people in the scene, a mechanism detects those that stop moving. Each moving region identified in scene is represented by their streakflows and location, modelled using Gaussian Mixture Models (GMMs). The inter-dependent relationships between their movements and locations, respectively, are modelled both at a given location point and by their dynamics (over a certain time period). The changes in inter-relationship differences over time are modelled using Kernel Density Estimation (KDE) and finally the activity sequences are classified into group activity classes using Support Vector Machines (SVM). Each stage of the proposed methodology is discussed in more detail in the forthcoming sections.

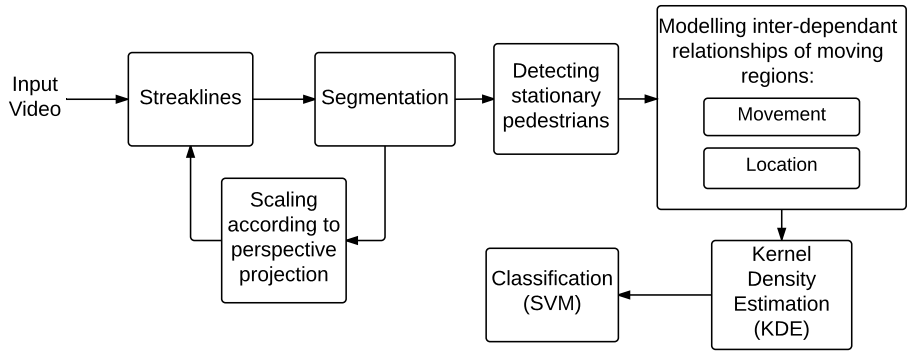


Figure 5.1: Overview of the proposed group activity recognition approach

The first processing stage consists of movement estimation via streakflow. As previously discussed extensively in Chapter 4, streakflows correspond to tracking fluid particles that have passed through a particular location in the past and their modelling is based on the Lagrangian framework for fluid dynamics [65]. The streakflows represent the fluid like flow in a scene, enabling the filling of spatial gaps. Similarly to the approach applied to crowded scenes in Chapter 4, small gaps are a common occurrence in group scenes, particularly when pedestrians are passing one another and are briefly occluded. The

streakflows are initialised as described in Chapter 4, by using a grid of particles which are then moved along using the dense optical flow. The streakflow is then computed as described in Chapter 4, assuming pixel similarity in eight-connectivity neighbourhoods. One difference between the streakflow approach in Chapter 4 and the approach to group activity is that we fit a first degree polynomial to the streakline in order to obtain a smoothed representation of the streakline, where a single vector now represents the flow over several frames.

One further issue with the earlier streaklines approach from Chapter 4 was that the motion consistency over several frames was not considered. This was not an issue with a single camera fixed on a particular scene, but when such methods are applied to real world environment containing either hand-held or PTZ cameras, such approaches become prone to noise from changes in camera parameters and camera movement.

In this approach, the consistency of the streaklines over several frames is considered in order to obtain a more robust estimation of the movement of individuals in the scene. Furthermore, poor estimations of movement will propagate errors through to the stationary pedestrian detector stage, potentially causing false positives and may degrade the models of the group activities. The motion consistency is checked by ensuring that the motion is present in the scene across several sets of frames and not just across a single set of frames. More specifically, motion must be present in the same region at least a certain percentage of frames over a particular time, otherwise the motion is considered noise and is subsequently removed. The percentage of frames defining the continuation of the movement is defined empirically, as described in the experimental results section, and is dependant on the amount of variation expected in human movement and the type of scene under observation.

Similarly to Chapter 4, we make the assumption that each compact region of streakflows may contain several individual movements, which can be represented by clusters. The Expectation-Maximization (EM) algorithm, under the Gaussian Mixture Model (GMM) modelling assumption is used for segmenting and modelling each inter-connected region as described in detail in Chapter 4. The space of clustering is defined jointly by both movement and localisation, as given by the streakflows and their locations in the frame, respectively.

One common issue with video sequences acquired with surveillance cameras is that the movement flow may be affected by perspective distortion of the scene. The perspective

distortion of the video sequences is often evident in the case of video sequences from video surveillance cameras where wide-angle lens cameras are used, often located at low to medium heights. Although the effects of perspective distortion are usually consistent over the entire video feeds due to the prevalent use of fixed cameras, modern surveillance systems often make use of pantiltzoom cameras (PTZ cameras) which have the ability to pan, tilt and zoom over a wide area leading to changes to the perspective projection modelling parameters in time. A dynamic perspective projection model system would be required in the case of PZT cameras. As is the case with most group activity modelling approaches from the literature, they require the manual annotation of the movement in the scene. To address this, our approach to perspective projection correction is much simpler using a two-step approach to movement segmentation using a dynamic scaling factor.

In the first step, the segmentation is performed in order to estimate the height of the moving objects, which is used to derive a scaling factor. More specifically, the segmentation is performed as described earlier in Chapter 4. The height of each moving region in a particular interconnected area of the scene is recorded. A scaling factor is computed for each moving region i as follows:

$$s_i = \frac{1}{2h_m}(h_i + \frac{\sum_{j=1}^{n} h_j}{n}) \tag{5.1}$$

where s_i is the scaling factor, for the moving region i. h_i is the height identified for each moving region in the first step, $j = 1, \ldots, n$ are the segmented moving regions in a given inter-connected region, h_m is the predetermined overall mean height of all moving regions. s_i is therefore considered to be the scaling factor between the average moving region and the regions in the particular inter-connected area of the scene. The other moving regions $j = 1, \ldots, n$ are used in the equation to add a robustness to the scaling factor by considering that all moving persons in a particular interconnected regions are of a reasonably similar height.

All movement streakflow vectors, defined by $\mathbf{M}_i$ for the region i are then scaled by the motion scaling factor s_i:

$$\mathbf{M}_i^{'} = s_i \mathbf{M}_i. \tag{5.2}$$

This is repeated for all compact moving areas which are identified in the scene. The flow vectors across all moving regions in the scene should now be scaled appropriately, and the segmentation is then reapplied using the newly scaled motion flows. Each moving region is finally represented by a GMM defined by its characteristic parameters defining

its movement and location in the scene as described in Chapter 4.

A further issue addressed in this study is the modelling of individuals who become stationary after they have moved through the scene. Unlike in manual tracklet models where stationary pedestrians would be annotated, under the proposed optical flow detection and motion model persons would not be accounted for in the scene, individually. To overcome this situation, we propose to identify when and where people stop moving in the scene without the use of manual pedestrian labelling or manually initialised tracklets. The proposed model is based on the principle that if a pedestrian who is moving in the scene stops, then they are deemed stationary and their motion and location parameters should be recorded until they begin moving again. More specifically, if no movement is present in a particular region where motion was previously detected during p consecutive frames, this indicates a person or a group of persons that are stationary at that moment. Such stationary regions are characterised by their location and by zero motion. When movement occurs again within a bounding box of the stopped pedestrian, the region is deemed to be no longer stationary and the new emerging moving region in the area is activated in the existing group activity model. Any movements of a person present near the edge of the scene that subsequently moves out of the scene is identified and the respective moving region is no longer considered. The robustness of the flow over several sets of frames as described earlier in this section largely eliminates the potential for false positives caused by any erroneous movement vectors in the scene such as from camera movement.

5.3 Modelling Interdependent Relationships of Moving Regions

The key characteristics of group activities are often present in the interdependent relationship between the pedestrians/moving objects. In this work, the interdependent relationships are modelled by pairing each two moving regions identified in the scene and by evaluating the features of their interdependencies. In this section, four distinct features are presented for representing human group interactions: streakflow differences, streakflow dynamics, location differences and location dynamics.

The interdependent relationships are calculated as relative differences in the movement and location spaces. For modelling the movement we consider, as in Chapter 4, streakflows. This aims to model the inter-dependant relationship of the movement for

a group of people at a particular time instance. For example, if individuals in a group exhibit similar streakflows in an activity such as running in a group, their probability density functions characterising their movement will be similar, and consequently differences in their streakflow estimates would be close to zero. If individuals in a group exhibit very different streakflows, for example as in the "Ignoring activity", resulting in large differences between the movement estimates of their component individual moving regions. Kulback-Leibler (KL) is used to calculate statistical interdependencies between pairs of moving regions found in the scene, as discussed and described in the previous chapter. KL divergence is a traditional statistical measure of the difference between two probability distributions. The generalised version of the KL divergence between streakflow model $\mathcal{A}_{I(t)}$ and streakflow model $\mathcal{A}_{J(t)}$, where $I(t)$ and $J(t)$ are two moving regions at time t is given by:

$$D_{KL}(\mathcal{A}_{I(t)}||\mathcal{A}_{J(t)}) = \int_{-\infty}^{\infty} p(\mathcal{A}_{I(t)}|x) \ln \left(\frac{p(\mathcal{A}_{I(t)}|x)}{p(\mathcal{A}_{J(t)}|x)} \right) dx \tag{5.3}$$

by assuming Gaussian mixture models (GMM) for the features characterising the moving regions from the scene, the KL divergence can be computed in closed form between streakflow model $\mathcal{A}_{I(t)}$ with mean vector $\boldsymbol{\mu}_{I(t)}$ and diagonal covariance matrix $\boldsymbol{\Sigma}_{I(t)}$ and streakflow model $\mathcal{A}_{J(t)}$ with mean vector $\boldsymbol{\mu}_{J(t)}$ and diagonal covariance matrix $\boldsymbol{\Sigma}_{J(t)}$, where $I(t)$ and $J(t)$ are the two moving regions at time t by [30]:

$$\begin{aligned} D_{KL}(\mathcal{A}_{I(t)}||\mathcal{A}_{J(t)}) = \; & 0.5 \Big[\log(\det(\boldsymbol{\Sigma}_{J(t)}) / \det(\boldsymbol{\Sigma}_{I(t)})) + \operatorname{tr}(\boldsymbol{\Sigma}_{J(t)}^{-1} \boldsymbol{\Sigma}_{I(t)}) \\ & + (\boldsymbol{\mu}_{J(t)} - \boldsymbol{\mu}_{I(t)})' \boldsymbol{\Sigma}_{J(t)}^{-1} (\boldsymbol{\mu}_{J(t)} - \boldsymbol{\mu}_{I(t)}) - d \Big] \end{aligned} \tag{5.4}$$

where d is the number of dimensions, and in the case of streakflow models or location models, $d = 2$. Equation 5.4 therefore provides a measure of the difference between the two probability distributions for streakflow models $\mathcal{A}_{I(t)}$ and $\mathcal{A}_{J(t)}$.

One downside of using the standard KL divergence from Equation 5.4 is that it is not symmetric. A symmetrised version of the KL divergence can be computed by:

$$D_{SKL}(\mathcal{A}_{I(t)}||\mathcal{A}_{J(t)}) = \frac{1}{2} \left[D_{KL}(\mathcal{A}_{I(t)})||\mathcal{A}_{J(t)}) + D_{KL}(\mathcal{A}_{J(t)}||\mathcal{A}_{I(t)}) \right] \tag{5.5}$$

where $D_{KL}(\mathcal{A}_{I(t)}||\mathcal{A}_{J(t)})$ is the KL divergence between the streakflow distribution of moving regions $I(t)$ and $J(t)$. Finally, the scaled differences between two streakflow models

$\mathcal{A}_{I(t)}$ and $\mathcal{A}_{J(t)}$ for moving regions $I(t)$ and $J(t)$ at time t can be computed by:

$$M(I(t), J(t)) = \mathrm{e}^{-\frac{D_{SKL}(\mathcal{A}_{I(t)}||\mathcal{A}_{J(t)})}{\sigma_m}} \tag{5.6}$$

where σ_m is a scaling factor for movement differences and $D_{SKL}(\mathcal{A}_{I(t)}||\mathcal{A}_{J(t)})$ is the symmetrised KL divergence between the streakline distribution of moving regions $I(t)$ and $J(t)$ at time t.

$M(I(t), J(t))$ results in a difference value scaled within the range $[0, 1]$ which models the difference between the two streakflow models, each characterising the movement of one region in the scene, associated to a moving person. For example, individuals moving in completely different directions will have $M(I(t), J(t)) = 0$ whilst individuals exhibiting similar movements (characterised by similar direction and speed) will have $M(I(t), J(t)) = 1$. The differences are computed by considering all pairs of moving regions in the scene at a particular time t by using Equation (5.6). The differences are then concatenated to form a vector representing the inter-dependant group relationship of the streakflows at a particular time t.

Whilst the streakflow differences are a good representation of movement interactions at a particular time instance, they fail to account for differences that may occur over the medium term. For example, in a gathering group activity, movement over the medium term may appear quite similar. On the other hand, movement in a fighting activity may vary considerably in the medium term. To address this, we also model the dynamic changes of moving regions over consecutive time intervals. To model the dynamic changes, we compute the differences between moving regions over sets of frames by computing the differences between all streakflow models at time t and all streakflow models at time $t+n$ in the given scene. The dynamic differences between two streakflow models $\mathcal{A}_{I(t)}$ and $\mathcal{A}_{J(t+n)}$ for moving regions $I(t)$ and $J(t+n)$ at time t and $t+n$ respectively, can be computed by:

$$M(I(t), J(t+n)) = \mathrm{e}^{-\frac{D_{SKL}(\mathcal{A}_{I(t)}||\mathcal{A}_{J(t+n)})}{\sigma_m}} \tag{5.7}$$

This is similar to equation (5.6), except that the interdependencies in movement are now calculated across the time, measuring the dynamics of movement in the scene. A vector of streakflow differences representing all the inter-dependant relationships of streakflow models between the time instances t and $t+n$ is then formed.

The distributions of relative locations for the people from the scene, both moving or

stationary, is modelled similarly by considering differences between the GMMs representing the spatial-location of the moving regions. When considering modelling the location of moving regions, the mean will approximate the centre of the region, whilst the variance will provide some characteristics of the size and shape of the region. For example, groups of persons who are close together will exhibit only small differences in location GMMs, while individuals far apart will exhibit large differences in location GMMs. Furthermore, large groups forming a single interconnected region will have different GMMs characteristics to smaller groups due to the difference in the variance in their GMM models. Given two location GMMs $\mathbf{C}_{I(t)}$ and $\mathbf{C}_{J(t)}$ for moving regions $I(t)$ and $J(t)$ at time t, the differences between their locations can be computed by:

$$D(I(t), J(t)) = \mathrm{e}^{-\frac{D_{SKL}(\mathcal{C}_{I(t)}||\mathcal{C}_{J(t)})}{\sigma_l}} \tag{5.8}$$

where σ_l represents the characteristic scale parameter for locations. $D(I(t), J(t))$ provides a value in the range $[0, 1]$ which is the difference between the two location models. For example, individuals characterised by moving regions $I(t)$ and $J(t)$ at time t, located far apart, will have $D(I(t), J(t)) = 0$ whilst individuals very close together will have $D(I(t), J(t)) = 1$. A vector representing all the inter-relationships of locations for the group activity at time t is then formed accordingly.

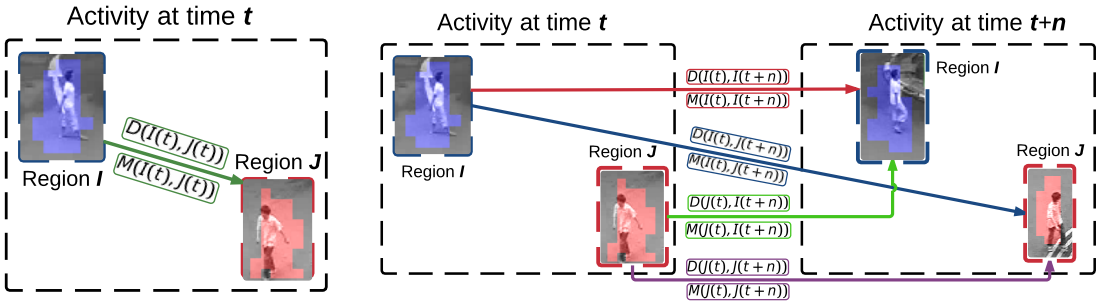
Similarly to the streakflow model, the dynamics of changes in movements' locations over time can also be computed. Unlike in the previously proposed static model, now the dynamics of relative movement and interaction within the group will be modelled. Changes in location over the medium term may be significant. For example, when individuals are performing a gathering group activity their locations tend to become closer together over time, but such a group relationship may not be evident from the differences modelled at single time instances.

Given two location GMMs $\mathbf{C}_{I(t)}$ and $\mathbf{C}_{J(t+n)}$ for moving regions $I(t)$ and $J(t+n)$ at time t and $t+n$, respectively, the differences between their locations can be computed by:

$$D(I(t), J(t+n)) = \mathrm{e}^{-\frac{D_{SKL}(\mathcal{C}_{I(t)}||\mathcal{C}_{J(t+n)})}{\sigma_l}} \tag{5.9}$$

The dynamic changes of differences are computed by the differences between each location of a centre of a moving region found at time t and any of those found at time $t+n$. using equation (5.9). A vector of location differences, representing all the inter-dependant relationships of location points between time t and $t+n$, is obtained. A visualisation of

both the base model and the dynamics model for both motion and location is shown in Figure 5.2.



a) Differences model b) Dynamic model

Figure 5.2: Modelling the inter-dependencies of moving regions in both space and time.

One further issue that arises when computing such differences is that the rate of movement change and rate of location change are not clearly characterised. For example, when using differences alone, the differences between the movements of individual people, both in time and space, taking part in the activities of running or walking may at first appear quite similar. To overcome this, we consider the background as an additional region for both the streakflow model and the location model. In the motion case, the background object is defined as the GMM model comprising of all the motion in the scene that does not belong to a moving region (often zero motion if the camera is stationary). In the latter case, the location object is defined as the GMM representing the centre of the scene. By adding the background model, the change in both motion and location relative to the background is characterised, representing the absolute movement of people in the scene. In the case of any camera movement, such a model would account for this. Given a streakflow background model $\mathcal{A}_{B(t)}$, at time t the difference between the streakflow model $\mathcal{A}_{I(t)}$, for moving region $I(t)$, at time t, and the background $B(t)$ is computed as:

$$M(I(t), B(t)) = \mathrm{e}^{-\frac{D_{SKL}(\mathcal{A}_{I(t)} || \mathcal{A}_{B(t)})}{\sigma_m}} \tag{5.10}$$

Similarly, given the centre point $\mathbf{C}_{B(t)}$ defined as the location of background model $B(t)$ (centre of the scene) at time t and the location model $\mathbf{C}_{I(t)}$ for moving region $I(t)$ at time t, the difference is computed as:

$$D(I(t), B(t)) = \mathrm{e}^{-\frac{D_{SKL}(\mathcal{C}_{I(t)} || \mathcal{C}_{B(t)})}{\sigma_l}} \tag{5.11}$$

Such differences are then computed between every region in the scene and the background model $B(t)$. Finally, the vector of differences in both cases are concatenated with the

vector representing pairwise motion and location differences between the moving regions in the scene.

5.4 Model Representation via Kernel Density Estimation

To model the change in feature relationships over the whole sequence, we propose to use bi-variate Kernel Density Estimation (KDE), computed over the difference over time. KDE will provide smoothing of the dynamics of feature changes over time increasing the robustness of the group activity model. Firstly, we form two column matrices where the motion or location inter-dependences for each pair of moving regions are represented along the first column and their corresponding time instances are located along the second column. This matrix representation is used for each feature (streakflow, streakflow dynamics, locations and location dynamics), separately. Therefore each video sequence will be represented by four two column feature matrices, where the feature is placed along one column and the time is along the second column.

In this work, we propose to use the bi-variate KDE method proposed in [12] which is based on using linear diffusion processes. In [12] they proposed an estimator which built on existing ideas for adaptive smoothing by incorporating information from a pilot density estimate. The KDE methodology from [12] assumes the kernel to be Gaussian and uses an automatic method for selecting the appropriate bandwidth for the given data. The use of KDE over traditional histograms has several key advantages, most notably adaptive smoothing of the data which not only helps with the smoothing of noise but provides smooth transitions between the models of the group activity features in time. Secondly, the automatic bandwidth selection method allows for different granularity of different features to be represented depending on the feature data. For example, some activities may exhibit very small changes in feature differences over time whilst some may have only large, well pronounced changes.

Using the bi-variate kernel density estimator, the data is sampled over a fixed grid size of $K \times K$, given the normalized matrix data discussed above. A visual representation of the matrix and KDE is shown in Figure 5.3. By using a fixed grid size, video sequences of different lengths will be normalized in length. This helps to normalise the difference in speeds at which the activities are performed. For example, a group gathering slowly would be normalized and appear similar to a group gathering at a much faster pace. The grid size is a important parameter in the density estimation as too small a grid would

result in over-smoothed feature data and consequently important characteristics in the relationship features may be lost. If the grid size is too large, then the data will appear too sparse and the KDE would not model well the underlying pattern of the data. The kernel for density estimation is assumed to be Gaussian. The bandwidth parameters of the bi-variate Gaussian kernel are used to help control the smoothing effects of the kernel density estimator.

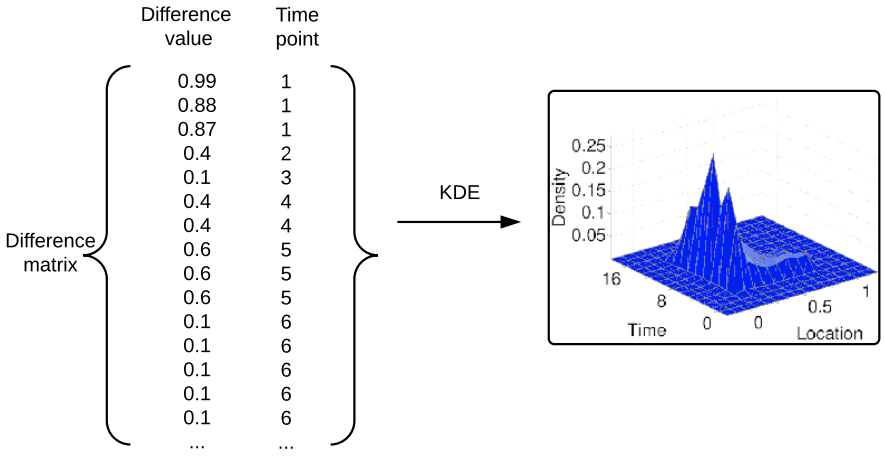


Figure 5.3: Example of the matrix representation and application of KDE

The densities computed over the fixed grid are used as the defining feature vector representation for the group activity. Such densities are computed independently for each feature, representing the relationships of the moving regions in the movement, movement dynamics, location and location dynamics, respectively. Finally, the feature vectors representing each activities are used to train a Support Vector Machine (SVM).

5.5 Experimental Results

The proposed approach has been evaluated on two state of the art group activity datasets - the NUS-HGA dataset [75] and the new Collective dataset [22]. In both datasets, only a single group activity is performed at any one time and sometimes, certain persons, who are not part of the group activity are crossing the scene. In [75] the activities are pre-segmented into separate video sequences whilst [22] contains video sequences where the activities flow from one activity to the next. Examples of activities from both datasets are shown in Figure 5.4. Both datasets contain perspective distortion to some degree, but the perspective distortion of the new Collective dataset shown in Figure 5.4 c) and d) is

significantly worse than of those in the NUS-HGA dataset in Figure 5.4 a) and b) due to the low camera angle. Furthermore, note that, due to the proximity of the recording camera and its settings, the pedestrians from the scene in the new Collective dataset generally appear much larger than those in the NUS-HGA dataset. The video sequences in the new Collective dataset are shot using a hand-held camera, therefore camera movement (and therefore motion noise) may be significant.

a) Fighting (NUS) b) Gather (NUS) c) Queuing (Col) d) Chasing (Col)

Figure 5.4: Example activities from the NUS-HGA [75] and new Collective datasets [22]

For all experiments in this chapter we follow the same recognition outline. To begin, the streakflows are extracted for each set of 10 frames and the moving regions are segmented based on the streakflows in each inter-connected region. The streakflow models and their respective location models are extracted for the moving regions identified in each set of frames. The features of the moving regions are then modelled by the inter-dependant differences between both the streakflow and location models between all moving regions across a set of frames. Similarly, the inter-dependant differences between both the streakflow and location models between two consecutive sets of frames (dynamic model) are also extracted. This is repeated for all sets of frames in the video sequences. Finally, the vector of (motion or location) differences for each video sequence are used to form a two column matrix with differences along the first column and the time instance along the second column. Bi-variate KDE is applied on a fixed grid size using the data from each motion/location feature matrix. The motion and location features are therefore represented by their probability density estimation (pdf) with difference in features along one axis and time along the other obtained from applying the KDE. Finally, the pdf's are used as the final features to feed a classifier and make recognition decisions via a Support Vector Machine (SVM) (with RBF kernel).

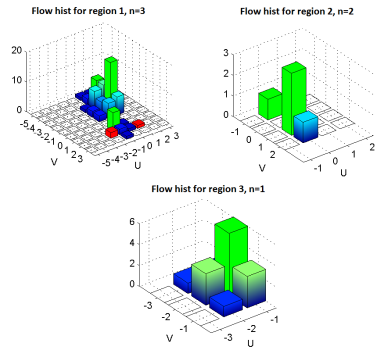
Experimental results on NUS-HGA dataset

The NUS-HGA dataset [75] consists of six different group activities collected in five different sessions, each session with different actors, different number of actors or at a different time of day. The resolution of the video is 720×576 at 25fps, and each activity sequence is approximately 4 to 10 seconds long. In total there are 476 video sequences across the 6 activity sequences. As shown in Figure 5.4 a) and b), the scene is outdoors. The scene varies somewhat by camera angle and lighting, and the tree visible in the centre of the scene often casts a shadow over the scene as observed in Figure 5.4 a).

To begin, streaklines are extracted as described in Section 5.2 for blocks of size 14×14 over 10 consecutive frames. The extraction of each set of streaklines are overlapped temporally by 3 frames, for example, over frames 1-10, then 4-14, etc. The motion filter described in Section 5.2 is placed over each 5 sets of frames, where motion must be present in at least 3 out of 5 sets of consecutive frames. Since the streaklines are overlapped temporally by 3 frames, the 5 sets of frames covers 22 frames whilst the the 3 sets of frames covers 18 consecutive frames. This process ensure motion consistency over several frames and aids in the removal of camera movement noise and in the removal of frivolous human movements such as minor hand movements. The motion histograms described in Section 5.2 are computed for each moving region and any entry in the histogram with a height below 15% of the maximum bar height is considered to be noise and is subsequently removed. This procedure ensures confidence by robustly estimating the human activity movement, based on significant movement, while removing the erroneously defined moving regions from further processing. The modes of the histograms are used as input to the EM algorithm and the segmentation is performed as described in Section 5.2. Each moving region is then represented by its streakflow mixture model and its location mixture model as described in Section 5.2.

Figure 5.5 displays an example of the streakflows, motion histograms and an example of the moving region segmentation for a fight activity from the NUS-HGA dataset. In this particular activity (fighting), movement is very intense and quite chaotic. In Figure 5.5 b) the solid green bars correspond to peaks of the histogram, while the solid red bars are entries which are removed due to their insignificance. In region 1 of Figure 5.5 c) - two to three different movements are present which are reflected in the histograms in Figure 5.5 b). Region 2 from Figure 5.5 contains a single dominant movement in the region although 2 peaks are detected in the histograms in Figure 5.5 b). This is due to some erroneous

a) Streakflows



b) Histograms of flow

c) Inter-connected regions

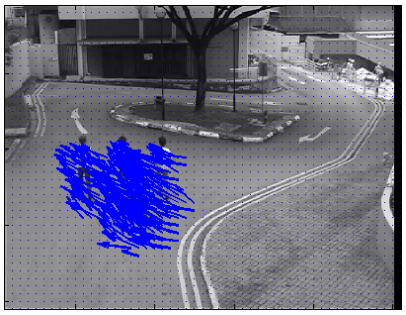
d) Moving regions after segmentation

Figure 5.5: Examples of streakflows, extracted from video sequences, showing group activities in a scene from NUS-HGA dataset. Note in b) n refers to the number of histogram peaks.

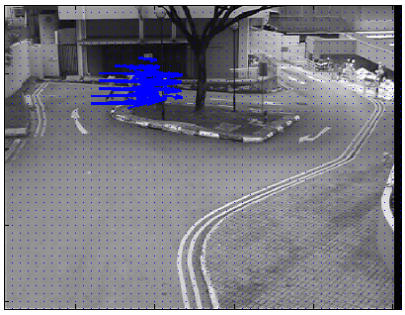
vectors being present. Despite this, the region is not included in the final segmentation in Figure 5.5 d) due to its insignificance. Region 3 from Figure 5.5 c) contains a single dominant movement as it can be observed that it is indicated by the histogram from Figure 5.5 b). This is reflected in Figure 5.5 d) where only a single moving region is segmented corresponding to region 3 from Figure 5.5 b). The small regions obtained in region 1 of Figure 5.5 d) help characterise the smaller atomic events performed in the group. Such movements are often lost when long term tracklets are used. Overall, the moving regions are well segmented and represent the movements of the humans very well.

Following the initial movement segmentation, the motion in each moving region is scaled according to the height of the region using equation (5.2). The segmentation is then performed for the second time using the scaled motion. Examples of the motion and segmentation after scaling is shown in Figure 5.6 for the run activity. Figure 5.6 shows an example of the start and end of a run activity sequence from the NUS-HGA dataset. In

Figure 5.6 a) and Figure 5.6 b) the intensity of the motion is consistent despite the evident effects of the perspective distortion, present in the representation of certain persons in the video sequence. The segmentation shown in Figure 5.6 c) and Figure 5.6 d) is consistent with each other and the two groups of individuals running are well segmented across both examples.

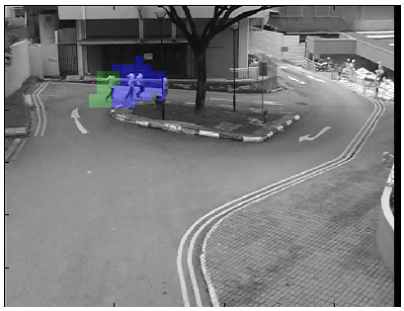


a) Streakflows at the start of the activity



b) Streakflows at the end of the activity

c) Segmentation at the start of the activity

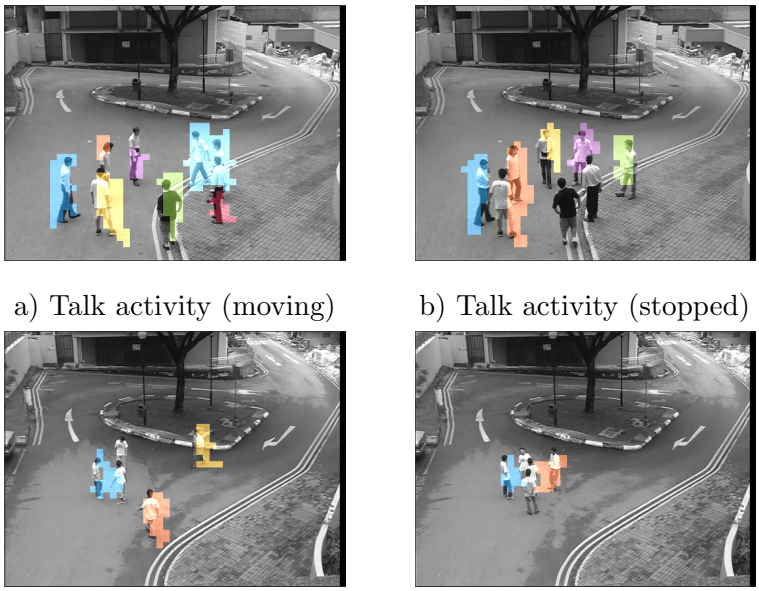


d) Segmentation at the end of the activity

Figure 5.6: Example of streakflows and segmentation at the start and end of a running activity sequence from the NUS-HGA dataset.

Following the second movement segmentation step, the stationary pedestrian detector is applied as described in Section 5.2 where the number of prior frames p is set to 25. We define the boundary parameter from Section 5.2 as 10% of the region size. Two examples of detecting stationary pedestrians are shown in Figure 5.7 for the talking and gathering activities. In Figure 5.7 a) and c) the pedestrians are still moving and therefore moving regions are detected. In Figure 5.7 b) and d) the individuals have stopped but their regions are still detected by the stationary pedestrian detector despite the fact that actually no motion is present in the scene at that instance.

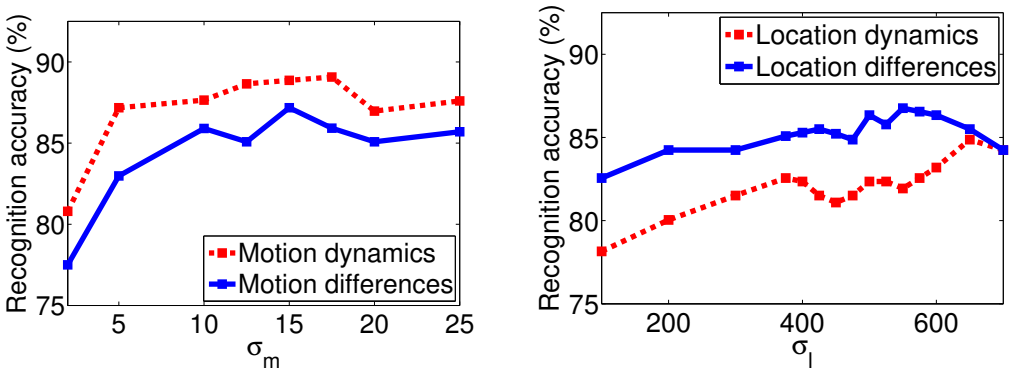
The next stage involves computing the streakflow differences, streakflow dynamics, location differences and location dynamics as described in Section 5.3. The size of the dynamic window for both streakflow dynamics and location dynamics is set to $n = 2$ sets



a) Talk activity (moving) b) Talk activity (stopped)

c) Gather activity (moving) d) Gathering activity (stopped)

Figure 5.7: Example of the stopped pedestrian detection when applied to gathering and talking activities from the NUS-HGA dataset. a) and c) show moving regions before stopping and b) and d) show the detected regions when the pedestrians are stationary.



a) Recognition accuracy as σ_m is varied b) Recognition accuracy as σ_l is varied

Figure 5.8: Recognition accuracy as the scaling parameters are varied for both streakflow and location features.

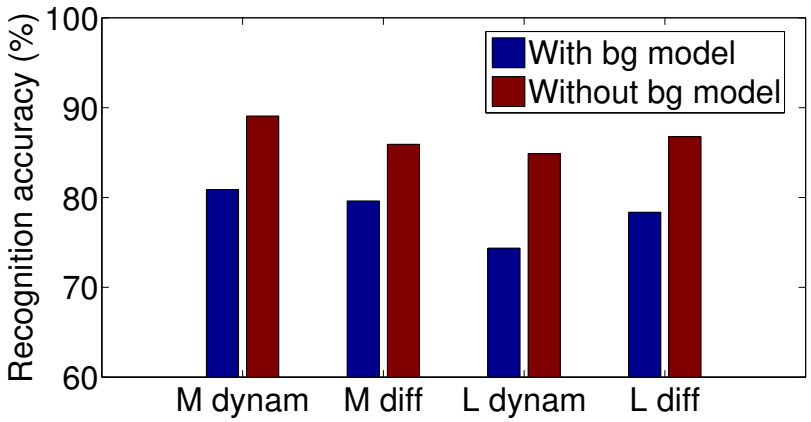


Figure 5.9: Difference in recognition accuracy when the background model is included.

of 10 frames each for the initial experiments. To begin, the scaling parameter σ_m is varied for the streakflow differences and streakflow dynamics. Figure 5.8 a) shows the difference in recognition accuracy as σ_m is varied for both the motion differences and the motion dynamics. From Figure 5.8 a) it is clear that the best recognition result is obtained when $\sigma_m = 15$ and $\sigma_m = 17.5$ for the streakflow differences and streakflow dynamics respectively. Similarly, the the scaling parameter σ_l is varied for the location differences and location dynamics. Figure 5.8 b) shows the difference in recognition accuracy as σ_l is varied for both the location differences and the location dynamics. From Figure 5.8 b) it is clear that the best recognition result is obtained when $\sigma_l = 550$ and $\sigma_l = 650$ for the location differences and location dynamics respectively. Notably in Figure 5.8, the recognition accuracy does not change significantly while the scaling parameters are varied. Therefore the selected parameters are $\sigma_m = 15$ and $\sigma_l = 550$ for streakflow differences and location differences, and $\sigma_m = 17.5$ and $\sigma_l = 650$ when streakflow dynamics and location dynamics are used.

In the following we add the background as one of moving regions as described in Section 5.3. Actually, in the video sequences analysed in here, the background represents the dominant region, characterised by zero motion. The recognition accuracy with and without the background model are shown in Figure 5.9. M dynam and M diff refer to the motion dynamics and motion differences, respectively; while L dynam and L diff refer to the location dynamics and location differences. A clear improvement in recognition results is shown across all features when the background model is included. This demonstrates the effectiveness of adding the background model to the differences and dynamics models.

The size of the dynamic window, which was set to $n = 2$ sets of frames in the previous

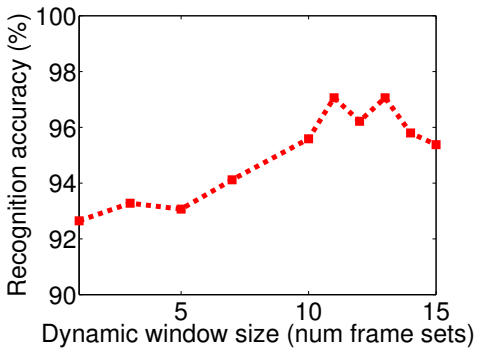


Figure 5.10: Recognition result as the size of the dynamic window n is varied.

experiments, is now varied between a single set of frames and $n = 15$ sets of 10 frames each. Figure 5.10 shows the recognition accuracy as n is varied for the combination of dynamic models (streakflow and location). The best recognition result is obtained when $n = 13$. Beyond $n = 13$, the result deteriorates as seen in Figure 5.10. It is suggested that when $n = 13$ over $n \ll 13$ the result improves due to there being no significant change in the motion/location features over only a few frames. As n increases, the motion/location features become sufficiently different from the starting frames features such that these better represent the dynamic changes in motion and location space. Therefore, we set $n = 13$ is used for the following.

Following the computation of the streakflow differences, streakflow dynamics, location differences and location dynamics, the data is represented over time using Kernel Density Estimation (KDE) as described in Section 5.4. The data is represented by a 2 column matrix over time as described in Section 5.4, where the feature is placed along one column and the time is along the second column. KDE is applied over a fixed grid size using the 2-column feature matrices as input data. The grid size parameter K is varied and compared to histograms of the same size. The results of K being varied and its histogram comparison is shown in Figure 5.11. Notably, $K = 16$ provides the best recognition results. In Figure 5.11, the KDE results shows a notable improvement over the equivalent histograms, demonstrating the effectiveness of KDE over histograms. In our experimental work, there was no improvement in recognition results by using grid sizes larger than $K = 16$. Furthermore, the computational complexity increases significantly when grid sizes larger than $K = 16$ are used. Therefore, in our experiments, $K = 16$ and the KDE approach is applied on the 2-column feature matrices as described above.

Examples of the density estimations and histograms for the motion differences are

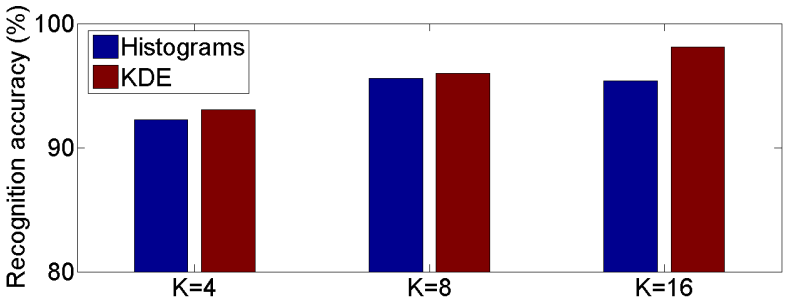
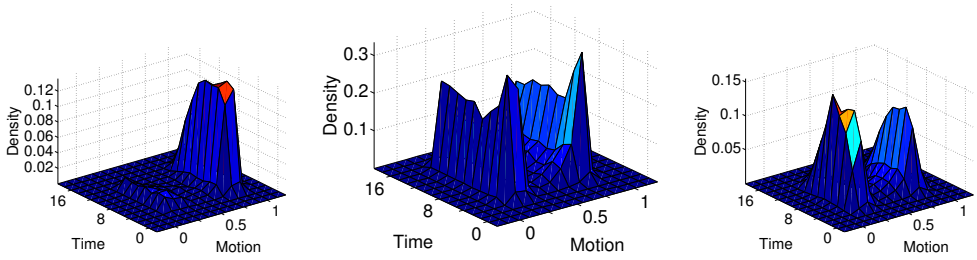
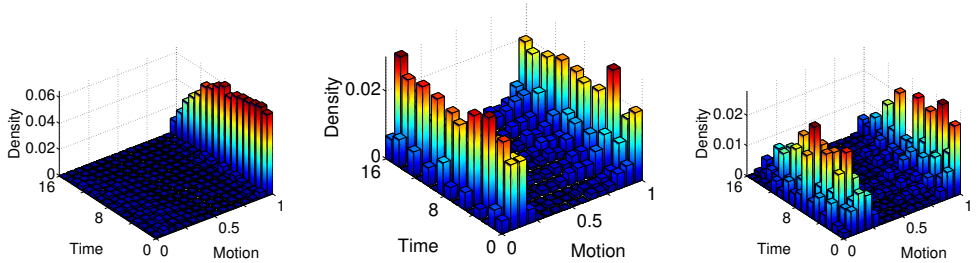


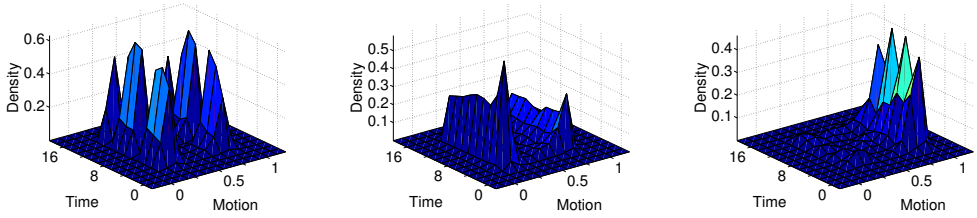
Figure 5.11: Recognition results as K is varied when using KDE and histograms.



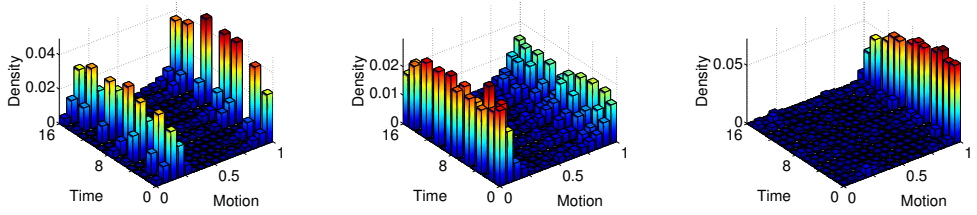
a) Walk (KDE) b) Ignore (KDE) c) Gather (KDE)



d) Walk (Hist) e) Ignore (Hist) f) Gather (Hist)

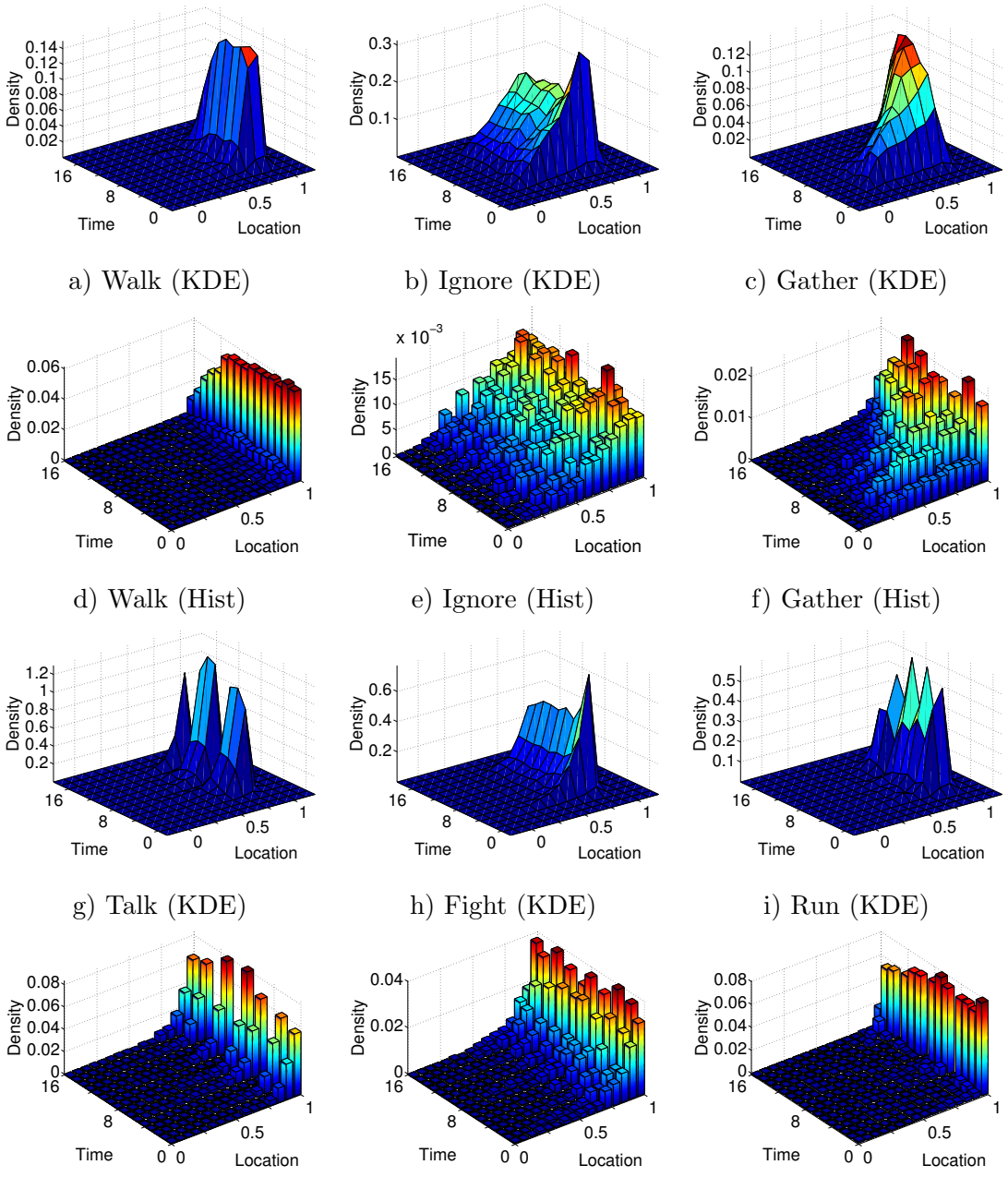


g) Talk (KDE) h) Fight (KDE) i) Run (KDE)



j) Talk (Hist) k) Fight (Hist) l) Run (Hist)

Figure 5.12: KDE and histograms representing the dynamics of the statistics of motion differences in time.



a) Walk (KDE) b) Ignore (KDE) c) Gather (KDE)

d) Walk (Hist) e) Ignore (Hist) f) Gather (Hist)

g) Talk (KDE) h) Fight (KDE) i) Run (KDE)

j) Talk (Hist) k) Fight (Hist) l) Run (Hist)

Figure 5.13: KDE and histograms for the dynamics of the statistics of relative positions of moving regions with respect to each other.

shown in Figure 5.12. The KDE plots show a clear smoothing effect in comparison to the histograms. For the run and walk activities in Figure 5.12, the motion differences appear quite similar, with motion differences close to 1 (indicating very similar motion difference patterns in the case of walking and running activities). The ignoring activity shown in Figure 5.12, shows a balance between motion that is substantially different, indicated by a value of 0 in the movement difference function, and motion that is similar, indicated by a value of 1 in the difference function. This is expected as the ignoring activity consists of individuals moving in the same direction and individuals moving in very different directions (ignoring each other). Similarly, the fight activity contains a balance between motion that is substantially different and motion that is similar. Notably, the density of the movement in the fighting activity is stronger than those in the ignoring activity. The densities from the gathering activity also appear quite similar to those of the fighting and ignoring activities, this is expected as some individuals will be walking in a similar direction to gather whilst some will be walking in a different direction. Finally, the talk activity also exhibits similar motion densities, this is largely due to the arm swinging and stepping back and forth which is present in the talking video sequences.

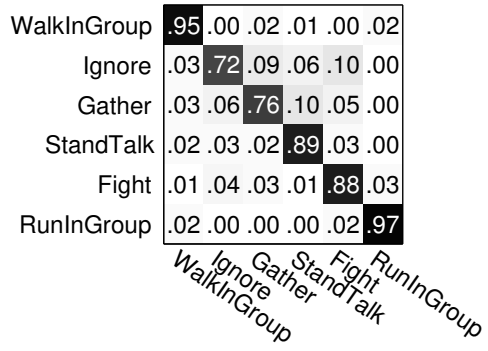
Examples of the density estimations and histograms for the location differences are shown in Figure 5.13. For the fighting activity in Figure 5.13 the location differences imply that the moving regions are all in close proximity. Similarly for the talking activity, the location differences in Figure 5.13 imply that the moving regions are standing very close together. The location differences for the gathering activity in Figure 5.13 shows a smooth transition from locations far apart (before gathering) at the start to locations close together at the end (gathered together). In the histogram representation of the gathering activity, the transition is still present, although it does not have the smooth transition present in the KDE plot. The location differences for the ignoring activity is well spread between large and small differences, this is because individuals are well spread and constantly moving around. The location differences for the walk and run activity are similar, with the locations appearing similar, this is expected as the walk and run activities contain groups of individuals running together in compact groups.

Notably, the density estimations in Figure 5.12 and Figure 5.13 for the corresponding activities are complementary. For example, while the fight, ignore and gather activities appear quite similar from motion differences in Figure 5.12, their corresponding location differences in Figure 5.12 are different. Such complementary features are extremely useful

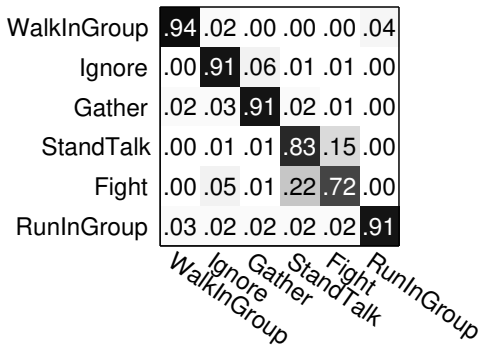
when the features are combined to build a more discriminative model.

For classification purposes, the density estimations for the different features are classified independently, and then combined to form an discriminant model as the motion and location features are often complimentary as previously observed in Figure 5.12 and Figure 5.13. For classification purposes, SVM are used with RBF kernel with parameters $C = 2.8284$ and $\gamma = 0.0019531$. For all experiments, we follow the evaluation protocol described in [75], where the NUS-HGA dataset is split into 5-fold training and testing and the performance is evaluated by average classification accuracy. Firstly, the four features (motion differences, motion dynamics, location differences and location dynamics) are used independently. Recognition results of the activities are shown through confusion matrices shown in Figure 5.14. The results of the dynamic features are notably better than the differences features. The confusion matrices show that the motion features poorly represent activities that are well represented by the location features and vice versa. This again shows the complimentary nature of the motion and location features. Notably, the recognition result for the gathering activity is improved when the dynamic motion features are sued over the motion differences. Similarly, the results of the talking activity are improved when the dynamic location features are used over the location differences. The motion and location differences are combined, and the results are shown in Figure 5.15 a). Similarly, the motion and location dynamics are combined and the results are shown in Figure 5.15 b). In both cases, the results are notably improved by combining the motion and location features. Finally, the combination of all four features (motion differences, motion dynamics, location differences and location dynamics) are shown in Figure 5.16. In this case, there is a notable improved over the results in Figure 5.15 and an even greater improvement when compared to the results of the individual features in Figure 5.14.

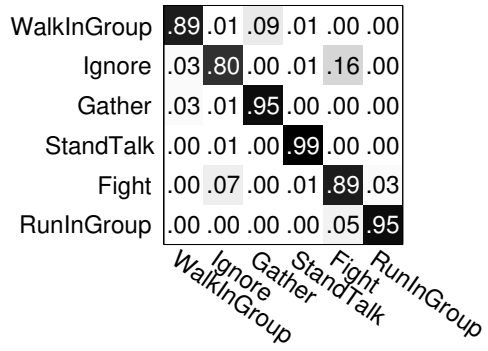
Comparisons of the results when compared to the state of the art are shown in Table 5.1. The location features provide a better recognition result than the motion features while the difference between the differences model for motion and location and the dynamics of motion and locations are quite significant. The combination of all features provides the best overall result (98%). Note that the group interaction zone method [21] does not evaluate the method using the 5-fold training and testing as suggested in [75], therefore slightly different results are expected from their method. In comparison to state of the art, we achieve a clear improvement in results (+2%) despite using an automated method unlike the other methods which all require manual annotation of tracklets or some form



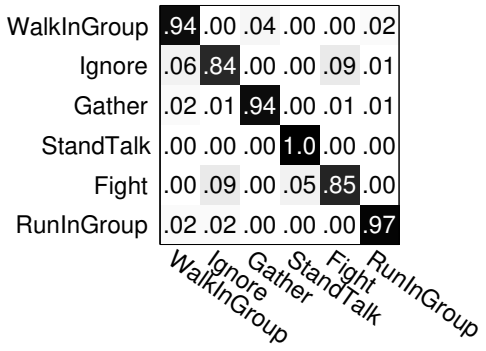
a) Motion differences - 86.16%



b) Location differences - 87.10%

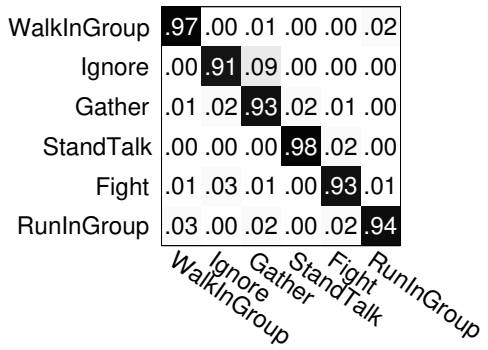


a) Motion dynamics - 91.59%

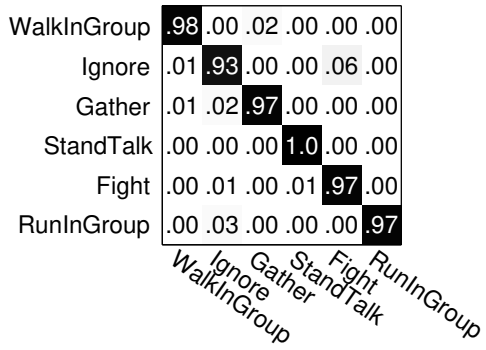


b) Location dynamics - 92.64%

Figure 5.14: Confusion matrices showing the recognition results of the four features on the NUS-HGA dataset.



a) Motion and location differences - 94.50%



b) Motion and location dynamics - 97.07%

Figure 5.15: Confusion matrices showing the recognition results when the motion and location features are combined when applied to the NUS-HGA dataset.

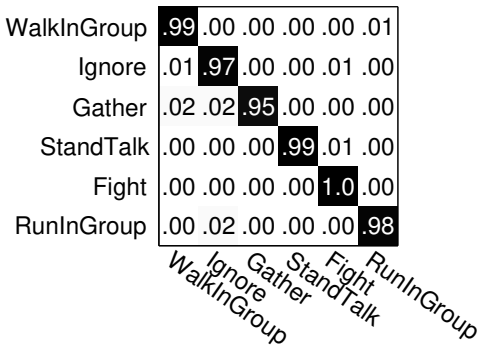


Figure 5.16: Confusion matrices showing the recognition results when the combination of all four features are used - 98%

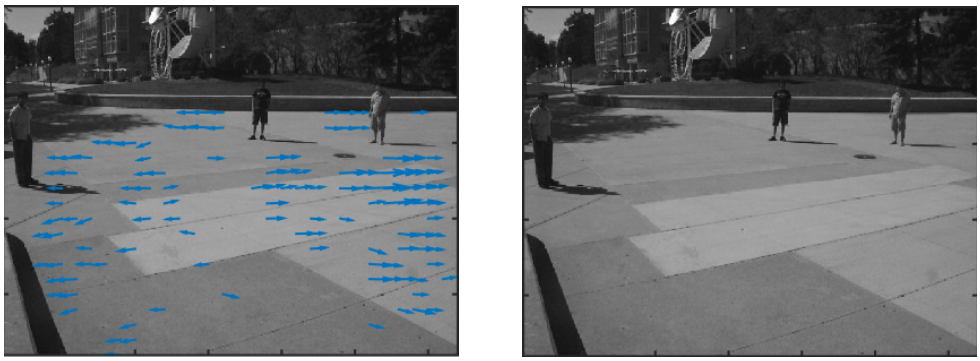
Table 5.1: Recognition results on the NUS-HGA dataset

Method	Result (%)
Localized Causalities [75]	74.2%
Group interaction zone [21]	96.0%
Multiple-layered model [20]	96.2%
Motion differences	86.2%
Location differences	87.1%
Motion dynamics	91.6%
Location dynamics	92.6%
Motion and location differences	94.5%
Motion and location dynamics	97.1%
Combined differences and dynamics	98.0%

of pedestrian detection training.

New Collective dataset

The new Collective dataset [22] consists of 32 video sequences with 6 collective activities: gathering, talking, dismissal, walking together, chasing and queueing. Each video sequence contains multiple instances of activities performed in an unspecified order. The video sequences are recorded using hand-held camera kept at low height, relative to the scene, when recording. Consequently, the resulting video recording is distorted by the perspective projection effects besides the camera noise.



(a) Streakflow with camera noise (b) Streakflow with motion filter applied

Figure 5.17: Example of the application of the motion filter on the new Collective dataset

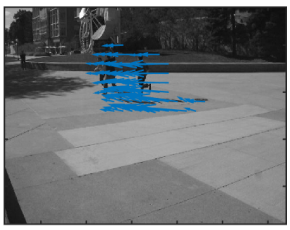
To begin, streaklines are extracted for blocks of size 20×20 pixels over 10 consecutive frames and the streaklines are overlapped temporarily by 3 frames. The motion filter described in Section 5.2 is placed over each 3 set of 10 frames, where motion must be present all three sets of frames. An example of the application of the motion filter is shown in Figure 5.17. The motion noise in Figure 5.17 a) is caused by camera shaking while being manipulated by the person taking the recording. In Figure 5.17 b) it can be observed that noise was completely removed by our filter. The two-step movement segmentation is applied as described in Section 5.2. Figure 5.18 shows examples of the streakflows and movement segmentation for the chasing and gather activities. The streakflows capture the human movement well without any erroneous motion from the camera movement. In both cases, the moving regions are well segmented, particularly in the chasing example where the chaser and chasee are segmented separately despite forming one connected region moving in the same direction.

The stationary pedestrian detector is applied as described in Section 5.2 where the number of prior frames, used for defining the movement fluidity following the motion detection, p, is set at 25. We define the boundary parameter from Section 5.2 as 15% of the region size. Two examples of transitioning stationary pedestrians through different activities are shown in Figure 5.19. In Figure 5.19 a) the pedestrians are moving together for the gathering activity, while in Figure 5.19 b) the individuals have stopped. Note that in Figure 5.19 b) the individuals are still detected by the stationary pedestrian detector. Finally, in Figure 5.19 c) the individuals are moving again (performing the dispersing activity) and the stationary regions are no longer recorded while the new moving regions are detected.

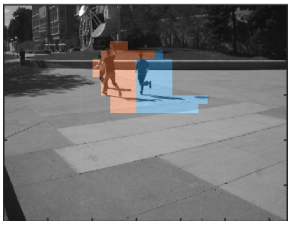
The next stage involves computing streakflow differences, streakflow dynamics, location differences and location dynamics as described in Section 5.3. The scaling parameters σ_m and $\sigma_l$are varied for both motion and location features respectively. Similarly to the NUS-HGA dataset, the best recognition results are obtained when $\sigma_m = 15$ and $\sigma_l = 450$ for both motion and location features respectively. The background model for both streakflows and locations are added to the features as previously described in Section 5.3. The size of the dynamic window n is set to $n = 5$. Unlike in the NUS-HGA dataset, each short of sequence of activity is of an unknown length and therefore may be quite short, limiting the size of the dynamic window n.

The features are represented over time by kernel density estimation as described in Section 5.4, where the parameter for the size of the fixed grid is set to $K = 8$. A small K would produce a very coarse representation of the movement or location dynamics, while a too large K would provide a noisy data model, lacking the ability to generalise.
To compare with the state-of-the-art, we follow the recommended evaluation protocol from [22] and divide the dataset into 3 subsets for 3-fold training and testing. Since the data sequences contain an unknown quantity of activities of an unknown length; we split the sequences during training by the start and end point of each activity, given by the groundtruth dataset. This is different to [22], where they split the video sequence into short sequences of a fixed length. This does not notably change the results as supervised learning is performed, therefore the label of the activity at any given time is given in the groundtruth of the dataset known during training. During testing, the sequences are split again by the start and end point of each activity and each short sequence is then evaluated in turn.

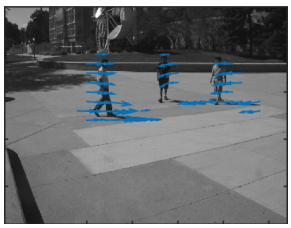
Confusion matrices of the results of our combined features compared to the approach from [19] are shown in Figure 5.20. One observation about our confusion matrix in Figure 5.20 is that the queuing activity is not well classified since pedestrians that are stationary and do not move for the duration are not well represented, whilst in manually annotated approaches the pedestrians are manually labelled from the beginning. Another observation from Figure 5.20 is that we achieve improved overall recognition results when the queuing activity is removed, and also greater consistency in the results across the other activities. The confusion matrix of [19], poorly recognising the gathering and chasing activities whilst our approach shows a clear improvement in recognising these activities. Comparison of our recognition results when compared to state of the art are shown in



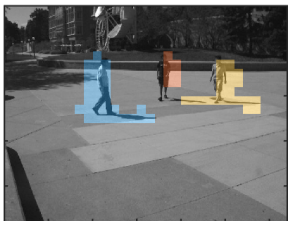
(a) Streakflow (Chasing activity)



(b) Segmentation (Chasing activity)

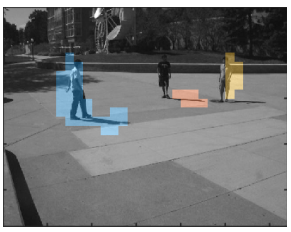


(c) Streakflow (Gather activity)

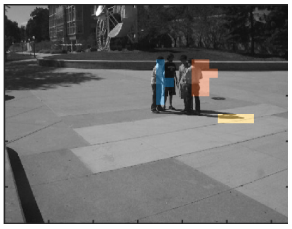


(d) Segmentation (Gather activity)

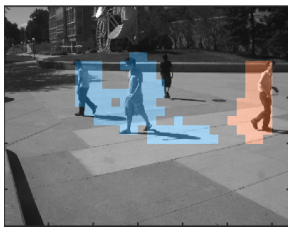
Figure 5.18: Examples of streakflow and segmentation on the new Collective dataset



a) Gathering



b) Talking



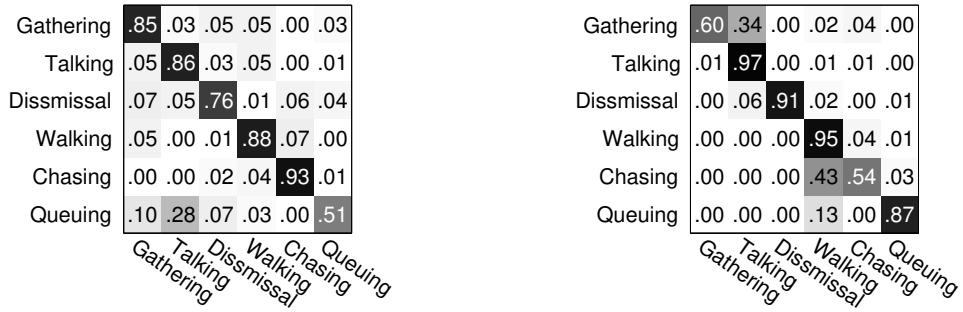
b) Dispersing

Figure 5.19: Example of transitions between activities in the new Collective dataset, including stopped pedestrian detection.

Table 5.2. The locations features outperform the motion features for both difference and dynamic features. The difference in results between the difference features and the dynamic features are not significant, although the dynamic features perform slightly better. Once again, the combination of motion and location features provide an improvement in the results, and the combination of all features provide the best overall result. Although our recognition result is slightly worse than that of [19], when the queuing activity is removed our results are significant better. This is expected, as the queuing activity relies heavily on the correct detection of the stopped pedestrians, and since the pedestrians are often close together while queuing, our method does not perform so well. Finally, we achieve such results without any complex pedestrian detection or manually annotated tracks.

Table 5.2: Recognition results on the new Collective dataset

Method	Result (%)
Monte Carlo Tree Search [5]	77.7%
Collective activities [23]	79.2%
MIR [19]	80.3%
Motion differences	68.4%
Location differences	70.1%
Motion dynamics	69.6%
Location dynamics	72.1%
Motion and location differences	76.5%
Motion and location dynamics	78.4%
Combined differences and dynamics	79.7%



(a) Motion, location and both dynamics - 79.7%

(b) Result from [19] - 80.3%

Figure 5.20: Confusion matrices for the recognition results on the new Collective dataset

5.6 Conclusions

In this chapter, we present an automatic approach for group activity recognition. We proposed a model to describe the discriminative characteristics of group activity by considering the relations between motion flows and locations of moving regions in the scene. Streakflows were used to represent the movement flows of individuals in the scene. Human movement was segmented using the EM algorithm under the Gaussian modelling assumption. Segmented moving regions were represented both in movement and location space by

their streakflow and location models. The interdependent differences of movements and locations were modelled using the symmetric KL divergence between the moving regions at a particular time instances. The dynamic differences between the moving regions were modelled similarly, by considering the interdependent differences of movements and locations at different time instances. We also proposed a scaling method using the height of regions as a scaling factor in order to compensate for the effect of perspective projection in video sequences with perspective distortion. In addition, we also proposed a stationary pedestrian detector to keep track of stationary pedestrians by marking the locations where they stop moving. Kernel Density Estimation (KDE) was used to model the interdependant differences over time, showing a clear improvement over using conventional histograms. Experimental results on the NUS-HGA dataset demonstrate the effectiveness of our approach, showing a 2% improvement over state of the art methods. Further experimental results on the new Collective dataset also demonstrates the effectiveness of our approach, showing competitive results compared to state of the art, without relying on any pedestrian detection or manual annotation of tracks like other methods.

Chapter 6

Conclusions

In this chapter, a summary of the contributions will be provided, discussing the strengths and weaknesses of the proposed activity recognition methodologies described in Chapters 3 to 5. This chapter will conclude with a discussion on the future directions of the work.

6.1 Contributions

In this section, the key contributions of the thesis to the field of human activity recognition will be highlighted.

Graph Based Human Activity Recognition

In Chapter 3, a graph based methodology was proposed by modelling the human activity as feature-relationship graphs. In this work, spatio-temporal activity regions were extracted, and features were modelled as similarity graphs across space and time. In other graph based approaches to human activity recognition, limitations were placed on the features due to issues representing and comparing the complex feature graphs. To overcome the limitations of typical graph based methodologies, the Laplacian representation of the graph was used, providing a vector-based representation of the graph while maintaining its discriminative nature. A further distinction of the proposed method is that the relationship between features was modelled; in the typical approaches to human activity recognition using BoW, the contextual and relationship between features is often ignored. While the results did not match those of the state of the art; it is suggested that this approach is better suited to more complex activities such as human interactions and contextual group activities.

The main drawback with the proposed approach is that a fixed number of cuboids were extracted from each video sequence to ensure feature graphs of the same size. This leads to an issue where some activities will naturally contain many more features than others; which means some activities will be poorly represented by the proposed methodology. Furthermore, the activity features are only modelled by their inter-dependent relationships; therefore the feature vectors themselves are not well modelled.

Abnormal Activity Identification Using Streaklines

In Chapter 4, a new approach is presented for abnormal human activity identification in crowded scenes. In this approach, a new approach to modelling activity regions was introduced by modelling the medium term flow of distinct moving regions. In this approach, the original streaklines approach was extended to a block-based methodology; where streakline flows were segmented using the EM algorithm under the Gaussian modelling assumption. Segmented regions were then represented in the movement and location space by their block-based streakflow and location models. PCA was then utilised to project the principal streakline vector representing each moving region. The streakline representation was extended to a multi-vector approach where each block is represented by its magnitude and direction vectors in the polar coordinates space. Furthermore, a weighting factor was introduced to balance the contribution of the magnitude and direction vectors. A novel localisation methodology was introduced to account for the perspective distortion in the scenes by only comparing activities with each other inside a dynamic window. The size of the dynamic window was based on the magnitude of the motion vectors and approximate distance from the camera. A further distinction of this methodology is that the dictionary of activities was generated online, thus allowing for the methodology to be used in an online system; without requiring offline training like some approaches. The proposed methodology also achieved state of the art results for localising abnormal activities in crowded scenes.

One issue with the current approach is that complex activities in the scene will be ignored due to the block-based based streakline approach. For example, a large region may be categorised as a runner rather than a walker, but if the individual is waving for help then this would be ignored due to the granularity of the waving activity. Furthermore, the interactions between pedestrians is not modelled in this methodology. Such a task is highly important in abnormal activity recognition, where anomalous activities generally

involve some form of human to human interactions.

Group based Activity Recognition

In Chapter 5, a novel automatic approach to group activity recognition was introduced. In this work, a model was proposed to describe the discriminative characteristics of group activity by modelling the relationships between moving activity regions. Unlike other methods which rely on manual annotation of tracklets; this method made use of the streaklines grouping introduced in the previous chapter. In this work, the interdependent differences of movements and locations modelled using the symmetric KL divergence between the moving regions at a particular time instances. This differs from other works in the area which only model the differences between longer term tracklets, and not the differences in movement and space over the short to medium term. A novel scaling method was proposed using the height of the regions as a scaling factor to compensate for the perspective distortion. In addition, a new stationary pedestrian detector was proposed to keep track of the stationary pedestrians by marking the locations when the pedestrians stop moving. In addition to modelling the differences in movement and location over time, the changes in such movement and location differences where also modelled using Kernel Density Estimation (KDE). The use of KDE showed a clear improvement over using conventional histograms. This differs from other methods which usually only consider the differences in features at a particular time, and do not model the changes in such differences over time. Experimental results on state of the art group activity datasets show a clear improvement of 2% over state of the art methodologies.

One drawback of the proposed method is that without manually tracking the pedestrians, it may become difficult to track stationary pedestrians when the scenes are more complex. For example, when two or more pedestrians stop in a nearby area, the current method may not prove sufficient in determining which pedestrians have stopped. A further drawback of the proposed approach is that without long term tracks, the long term movement and spatial changes are not well modelled by the method. This may become an issue when group activities are performed over a longer time period, and where the distinguishing characteristics are only present in the long term tracklets.

6.2 Future Work

This thesis made several novel contributions to the field of human activity recognition. Given these contributions, it is important that the works continues to improve and develop to provide further applications in human activity recognition and advance the field of computer vision further.

It is suggested that the graph based methodology introduced in Chapter 3 could be improved by removing the limitation of the fixed number of cuboids/feature vectors per activity graph. This would allow activities that have more motion/activity present to have more features, while activities with fewer movements can be represented by fewer features. A second suggestion could be to adopt a grid based approach by modelling the features as graphs for each region in the grid. This could prove useful to provide a better model of more localised activities. A further suggestion could be to model the features directly as a feature graph as opposed to modelling their differences, and compare such graphs using a graph matching methodology. This would allow features to be compared directly rather than through a graph embedding approach.

While the streakline-based anomaly detection approach introduced in Chapter 4 produced state of the art results for activity localisation, the methodology can still be improved further. One suggestion is to use a more robust method of segmenting activity regions rather than the heuristic histogram approach adopted thus far. This should provide improved activity segmentation and therefore improve activity recognition results. A further suggestion could be to improve the current method of generating the activity dictionary. At present, the activity models are only compared statistically to distinguish activity classes when a more robust method of determining initial activity classes could be introduced to improve the overall robustness of the activity dictionary. A further suggestion is to introduce new motion or appearance features into the feature pipeline to improve the saliency activity model. Furthermore, such new features could also be fused with other complimentary features, for example, motion and appearance features.

The group activity recognition approach introduced in Chapter 5 could be improved by fusing the current approach with a long term pedestrian tracking approach. By adding the tracking approach, the stationary pedestrians can be detected more robustly and the long term changes in motion and location would also be modelled. A further suggestion is to modify the proposed scaling methodology either by using a pedestrian detector (from the long term tracking method) or by some depth estimation to determine the approximate

distance of the pedestrians from the camera.

References

[1] S Abdelhedi, A Wali, and A M Alimi. Human activity recognition based on mid-level representations in video surveillance applications. *International Joint Conference on Neural Networks*, pages 3984–3989, 2016.

[2] A Adam, E Rivlin, I Shimshoni, and D Reinitz. Robust real-time unusual event detection using multiple fixed-location monitors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30:555–60, 3 2008.

[3] M A R Ahad, T Ogata, J K Tan, H S Kim, and S Ishikaw. Motion recognition approach to solve overwriting in complex actions. *IEEE International Conference on Automatic Face & Gesture Recognition*, pages 1–6, 2008.

[4] S Ali and M Shah. Human action recognition in videos using kinematic features and multiple instance learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:288–303, 2 2010.

[5] M R Amer, S Todorovic, A Fern, and Song-Chun Zhu. Monte Carlo Tree Search for Scheduling Activity Recognition. *IEEE International Conference on Computer Vision*, pages 1353–1360, 2013.

[6] M Baktashmotlagh, M T Harandi, and A Bigdeli. Non-Linear Stationary Subspace Analysis with Application to Video Classification. *International Conference on Machine Learning*, pages 450–458, 2013.

[7] A Basharat, A Gritai, and M Shah. Learning object motion patterns for anomaly detection and improved object detection. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[8] A F Bobick and J W Davis. The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:257–267, 2001.

[9] A F Bobick and A D Wilson. A state-based approach to the representation and recognition of gesture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:1325–1337, 1997.

[10] E Borzeshi, R Xu, and M Piccardi. Automatic human action recognition in videos by graph embedding. *International Conference on Image Analysis and Processing*, pages 19–28, 2011.

[11] E Z Borzeshi, O P Concha, and M Piccardi. Human action recognition in video by fusion of structural and spatio-temporal features. *Structural, Syntactic, and Statistical Pattern Recognition*, pages 474–482, 2012.

[12] Z I Botev, J F Grotowski, and D P Kroese. Kernel density estimation via diffusion. *Annals of Statistics*, 38:2916–2957, 11 2010.

[13] M Brand, N Oliver, and A Pentland. Coupled hidden Markov models for complex action recognition. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 994–999, 1997.

[14] M. Bregonzio, S Gong, and T Xiang. Recognising action as clouds of space-time interest points. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1948–1955, 2009.

[15] W Brendel and S Todorovic. Learning spatiotemporal graphs of human activities. *IEEE International Conference on Computer Vision*, pages 778–785, 2011.

[16] L W Campbell and A F Bobick. Recognition of human body motion using phase space constraints. *IEEE International Conference on Computer Vision*, pages 624–630, 1995.

[17] O Celiktutan, C Wolf, and B Sankur. Fast exact matching and correspondence with hyper-graphs on spatio-temporal data. *Journal of Mathematical Imaging and Vision*, 51:1–21, 1 2012.

[18] M Chang and W Ge. Probabilistic Group-Level Motion Analysis and Scenario Recognition. *IEEE International Conference on Computer Vision*, pages 747–754, 2011.

[19] X Chang, W Zheng, and J Zhang. Learning Person-Person Interaction in Collective Activity Recognition. *IEEE Transactions on Image Processing*, 24:1905–1918, 6 2015.

[20] Z Cheng, L Qin, Q Huang, S Yan, and Q Tian. Recognizing human group action by layered model with multiple cues. *Neurocomputing*, 136:124–135, 7 2014.

[21] N Cho, Y Kim, U Park, J Park, and S Lee. Group Activity Recognition with Group Interaction Zone Based on Relative Distance Between Human Objects. *International Journal of Pattern Recognition and Artificial Intelligence*, 29:1555007, 3 2015.

[22] W Choi and S Savarese. A unified framework for multi-target tracking and collective activity recognition. *European Conference on Computer Vision*, pages 215–230, 2012.

[23] W Choi and S Savarese. Understanding collective activities of people from videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36:1242–1257, 6 2014.

[24] Y Cong, J Yuan, and J Liu. Sparse reconstruction cost for abnormal event detection. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3449–3456, 2011.

[25] X Cui, Q Liu, M Gao, and D Metaxas. Abnormal detection using interaction energy potentials. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3161–3167, 2011.

[26] P Dai, H Di, L Dong, L Tao, and G Xu. Group interaction analysis in dynamic context. *IEEE Transactions on Systems, Man and Cybernetics*, 39:34–42, 2 2009.

[27] T Darrell and A Pentland. Space-time gestures. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 335–340, 1993.

[28] S K Dhar, M M Hasan, and S A Chowdhury. Human activity recognition based on gaussian mixture model and directive local binary pattern. *International Conference on Electrical, Computer Telecommunication Engineering*, pages 1–4, 2016.

[29] P Dollar, V Rabaud, G Cottrell, and S Belongie. Behavior Recognition via Sparse Spatio-Temporal Features. *IEEE International Workshop on Visual Surveillance and Performance*, pages 65–72, 2005.

[30] J Duchi. Derivations for Linear Algebra and Optimization. Technical report, University of California, Berkeley, 2007.

[31] A A Efros and A C Berg. Recognizing action at a distance. *IEEE Conference on Computer Vision*, pages 726–733, 2003.

[32] Z B Ehsan, P Massimo, and X Y Da. A Discriminative Prototype Selection Approach for Graph Embedding in Human Action Recognition. *IEEE International Conference on Computer Vision Workshops*, pages 1295–1301, 2011.

[33] D Emms, R C Wilson, and E Hancock. Graph embedding using quantum commute times. *International Workshop on Graph-Based Representations in Pattern Recognition*, pages 371–382, 2007.

[34] S Escalera. Human behavior analysis from depth maps. *International Conference on Articulated Motion and Deformable Objects*, pages 282–292, 2012.

[35] Z Fan, G Li, L Haixian, G Shu, and L Jinkui. Star skeleton for human behavior recognition. *International Conference on Audio, Language and Image Processing*, pages 1046–1050, 7 2012.

[36] Adrien Gaidon, Zaid Harchaoui, and Cordelia Schmid. Temporal Localization of Actions with Actoms. *IEEE Transactions on Pattern Analysis and Machine Intelligenc*, 35:2782–2795, 11 2012.

[37] G Gao and S Sun. Trajectory-based human activity recognition using Hidden Conditional Random Fields. *International Conference on Machine Learning and Cybernetics*, pages 1091–1097, 2012.

[38] D M Gavrila. The Visual Analysis of Human Movement: A Survey. *Computer Vision and Image Understanding*, 73:82–98, 1 1999.

[39] S Gong and T Xiang. Recognition of group activities using dynamic probabilistic networks. *IEEE International Conference on Computer Vision*, pages 742–749, 2003.

[40] L Gorelick, M Blank, E Shechtman, M Irani, and R Basri. Actions as space-time shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29:2247–53, 12 2007.

[41] S Gudivada and A G Bors. Face Recognition using Ortho-Diffusion Bases. *European Signal Processing Conference*, pages 1578–1582, 2012.

[42] A Gupta and L Davis. Objects in Action: An Approach for Combining Action Understanding and Object Perception. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[43] E Indermühle, M Liwicki, and H Bunke. Recognition of handwritten historical documents: HMM-adaptation vs. writer specific training. *International Conference on Frontiers in Handwriting Recognition*, 2008.

[44] Y A Ivanov and A F Bobick. Recognition of visual activities and interactions by stochastic parsing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:852–872, 8 2000.

[45] F Jiang, J Yuan, S A Tsaftaris, and A K Katsaggelos. Anomalous video event detection using spatiotemporal context. *Computer Vision and Image Understanding*, 115:323–333, 3 2011.

[46] Y Jiang, Q Dai, X Xue, W Liu, and C Ngo. Trajectory-Based Modeling of Human Actions. *European Conference on Computer Vision*, pages 425–438, 2012.

[47] G Johansson. Visual motion perception. In *Scientific American Offprints*, pages 76–88, 1975.

[48] T Kadir, R Bowden, E J Ong, and A Zisserman. Minimal Training, Large Lexicon, Unconstrained Sign Language Recognition. *British Machine Vision Conference*, pages 1–96, 2004.

[49] Y Ke, R Sukthankar, and M Hebert. Spatio-temporal Shape and Flow Correlation for Action Recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[50] J Kim and K Grauman. Observe locally, infer globally: a space-time MRF for detecting abnormal activities with incremental updates. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2921–2928, 2009.

[51] T K Kim, S F Wong, and R Cipolla. Tensor canonical correlation analysis for action classification. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[52] L Kratz and K Nishino. Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1446–1453, 2009.

[53] S Kullback and R A Leibler. On Information and Sufficiency. *Annals of Mathematical Statistics*, 22:79–86, 3 1951.

[54] S S Kumar and M John. Human activity recognition using optical flow based feature set. *IEEE International Carnahan Conference on Security Technology*, pages 1–5, 2016.

[55] I Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(9):107–123, 2005.

[56] I Laptev and P Pérez. Retrieving actions in movies. *IEEE International Conference on Computer Vision*, pages 1–8, 2007.

[57] Y LeCun, B Boser, and J S Denker. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1:541–551, 10 1989.

[58] W Li, V Mahadevan, and N Vasconcelos. Anomaly detection and localization in crowded scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36:18–32, 1 2014.

[59] W Li, Q Yu, H Sawhney, and N Vasconcelos. Recognizing Activities via Bag of Words for Attribute Dynamics. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2587–2594, 2013.

[60] W Lin, H Chu, J Wu, N Sheng, and Z Chen. A heat-map-based algorithm for recognizing group activities in videos. *IEEE Transactions on Circuits and Systems for Video Technology*, 23:1980–1992, 11 2013.

[61] J Liu, S Ali, and M Shah. Recognizing human actions using multiple features. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[62] J Liu, J Luo, and M Shah. Recognizing realistic actions from videos "in the wild". *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1996–2003, 2009.

[63] R Lublinerman and N Ozay. Activity recognition from silhouettes using linear systems and model (in) validation techniques. *International Conference on Pattern Recognition*, pages 347–350, 2006.

[64] J Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:888–905, 8 2000.

[65] R Mehran, B E Moore, and M Shah. A streakline representation of flow in crowded scenes. *European Conference on Computer Vision*, pages 439–452, 2010.

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

[67] R Messing, C Pal, and H Kautz. Activity recognition using the velocity histories of tracked keypoints. *IEEE International Conference on Computer Vision*, pages 104–111, 2009.

[68] Shilpa Metkar and Sanjay Talbar. Motion Estimation Techniques for Digital Video Coding. In *Motion Estimation Techniques for Digital Video Coding*, SpringerBriefs in Applied Sciences and Technology, chapter 2. Springer India, 2013.

[69] D Moore and I Essa. Recognizing multitasked activities from video using stochastic context-free grammar. *American Association for Artifical Intelligence*, pages 770–776, 2002.

[70] D J Moore, I A Essa, and M H Hayes. Exploiting human actions and object context for recognition tasks. *IEEE International Conference on Computer Vision*, pages 80–86, 1999.

[71] M M Naeini, G Dutton, K Rothley, and G Mori. Action Recognition of Insects Using Spectral Clustering. *IAPR Conference on Machine Vision Applications*, pages 1–4, 2007.

[72] H Nallaivarothayan, C Fookes, S Denman, and S Sridharan. An MRF based abnormal event detection approach using motion and appearance features. *IEEE International Conference on Advanced Video and Signal-Based Surveillance*, pages 343–348, 2014.

[73] P Natarajan and R Nevatia. Coupled Hidden Semi Markov Models for Activity Recognition. *IEEE Workshop on Motion and Video Computing*, pages 10–10, 2007.

[74] A Y Ng, M I Jordan, and Y Weiss. On Spectral clustering: Analysis and an Algorithm. *International Conference on Neural Information Processing Systems: Natural and Synthetic*, pages 849–856, 2001.

[75] B Ni, S Yan, and A Kassim. Recognizing human group activities with localized causalities. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1470–1477, 2009.

[76] J Niebles, C W Chen, and L Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. *European Conference on Computer Vision*, pages 392–405, 2010.

[77] J C Niebles, H Wang, and L Fei-Fei. Unsupervised Learning of Human Action Categories Using Spatial-Temporal Words. *International Journal of Computer Vision*, 79:299–318, 3 2008.

[78] S Park and J K Aggarwal. Recognition of two-person interactions using a hierarchical Bayesian network. *International Workshop on Video Surveillance*, pages 65–76, 2003.

[79] S Park and J K Aggarwal. A hierarchical Bayesian network for event recognition of human actions and interactions. *Multimedia systems*, 10:164–179, 8 2004.

[80] X Peng, Y Qiao, Q Peng, X Qi, and P R Chengdu. Exploring Motion Boundary based Sampling and Spatial-Temporal Context Descriptors for Action Recognition. *British Machine Vision Conference*, pages 1–11, 2013.

[81] P Peursum, G West, and S Venkatesh. Combining image regions and human activity for indirect object recognition in indoor wide-angle views. *IEEE International Conference on Computer Vision*, pages 82–89, 2005.

[82] J Philbin and A Zisserman. Object mining using a matching graph on very large image collections. *Indian Conference on Computer Vision, Graphics & Image Processing*, pages 738–745, 2008.

[83] F Porikli and T Haga. Event Detection by Eigenvector Decomposition Using Object and Frame Features. *Conference on Computer Vision and Pattern Recognition Workshop*, pages 114–114, 2004.

[84] C Rao and M Shah. View-invariance in action recognition. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages II–316–II–322, 2001.

[85] K Rapantzikos. Dense saliency-based spatiotemporal feature points for action recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1454–1461, 2009.

[86] J Richiardi, S Achard, H Bunke, and D Ville. Machine Learning with Brain Graphs. *IEEE Signal Processing Magazine*, pages 58–70, 4 2013.

[87] M Rodriguez, J Ahmed, and M Shah. Action MACH a spatio-temporal Maximum Average Correlation Height filter for action recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[88] M S Ryoo and J K Aggarwal. Hierarchical recognition of human activities interacting with objects. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[89] M S Ryoo and J K Aggarwal. Semantic Representation and Recognition of Continued and Recursive Human Activities. *International Journal of Computer Vision*, 82:1–24, 11 2008.

[90] M S Ryoo and J K Aggarwal. Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. *International Conference on Computer Vision*, pages 1593–1600, 2009.

[91] S Samanta and B Chanda. FaSTIP: a new method for detection and description of space-time interest points for human activity classification. *Indian Conference on Computer Vision, Graphics and Image Processing*, pages 81–88, 2012.

[92] S Savarese, A DelPozo, and J C Niebles. Spatial-Temporal correlatons for unsupervised action classification. *IEEE Workshop on Motion and Video Computing*, pages 1–8, 2008.

[93] C Schuldt, I Laptev, and B Caputo. Recognizing human actions: A local SVM approach. *International Conference on Pattern Recognition*, pages 32–36, 2004.

[94] P Scovanner, S Ali, and M Shah. A 3-dimensional sift descriptor and its application to action recognition. *International Conference on Multimedia*, pages 357–360, 2007.

[95] M Shah. Learning human actions via information maximization. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[96] L Shao and R Mattivi. Feature detector and descriptor evaluation in human action recognition. *International Conference on Image and Video Retrieval*, pages 477–484, 2010.

[97] E Shechtman and M Irani. Space-time behavior-based correlation-or-how to tell if two underlying motion fields are similar without computing them? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29:2045–2056, 11 2007.

[98] Y Sheikh, M Sheikh, and M Shah. Exploring the space of a human action. *IEEE International Conference on Computer Vision*, pages 144–149, 2005.

[99] Y Shi, Y Gao, and R Wang. Real-time abnormal event detection in complicated scenes. *International Conference on Pattern Recognition*, pages 3653–3656, 2010.

[100] N T Siebel and S Maybank. Fusion of multiple tracking algorithms for robust people tracking. *European Conference on Computer Vision*, pages 1–15, 2006.

[101] R Souvenir and J Babbs. Learning the viewpoint manifold for action recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7, 2008.

[102] J Sun, X Wu, S Yan, L Cheong, T Chua, and J Li. Hierarchical spatio-temporal context modeling for action recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2004–2011, 2009.

[103] A Ta, C Wolf, G Lavoue, A Baskurt, A Guillaume, and L Baskurt. Recognizing and Localizing Individual Activities through Graph Matching. *IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 196–203, 2010.

[104] A Tamrakar, S Ali, Q Yu, and J Liu. Evaluation of low-level features and their combinations for complex event detection in open source videos. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3681–3688, 2012.

[105] M Thida, H Eng, and P Remagnino. Laplacian Eigenmap With Temporal Constraints for Local Abnormality Detection in Crowded Scenes. *IEEE Transactions on Cybernetics*, 6:2147–2156, 12 2013.

[106] S Todorovic. Human activities as stochastic Kronecker graphs. *European Conference on Computer Vision*, pages 130–143, 2012.

[107] C Tseng, J Chen, C Fang, and J Lien. Human action recognition based on graph-embedded spatio-temporal subspace. *Pattern Recognition*, 45:3611–3624, 10 2012.

[108] M Turk and A Pentland. Eigenfaces for Recognition. *Journal of Cognitive Neuroscience*, 3:71–86, 12 1991.

[109] A Veeraraghavan, A K Roy-Chowdhury, and R Chellappa. Matching shape sequences in video with applications in human movement analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:1896—1909, 12 2005.

[110] M Visontai. Detecting unusual activity in video. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 819–826, 2004.

[111] H Wang, A Kläser, C Schmid, and C L Liu. Dense trajectories and motion boundary descriptors for action recognition. *International Journal of Computer Vision*, 103:60–79, 5 2012.

[112] L Wang and D Suter. Informative shape representations for human action recognition. *International Conference on Pattern Recognition*, pages 1266–1269, 2006.

[113] L Wang, X Zhao, Y Si, L Cao, and Y Liu. Context-associative hierarchical memory model for human activity recognition and prediction. *IEEE Transactions on Multimedia*, 19:646–659, 3 2017.

[114] Y Wang, K Huang, and T Tan. Human Activity Recognition Based on R Transform. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[115] D Weinland, R Ronfard, and E Boyer. A survey of vision-based methods for action representation, segmentation and recognition. *Computer Vision and Image Understanding*, 115:224–241, 2 2011.

[116] Y Weiss. Segmentation using eigenvectors: a unifying view. *IEEE International Conference on Computer Vision*, pages 975–982, 1999.

[117] A Wiliem, V Madasu, W Boles, and P Yarlagadda. A Context Space Model for Detecting Anomalous Behaviour in Video Surveillance. *International Conference on Information Technology*, pages 18–24, 2012.

[118] G Willems, T Tuytelaars, and L V Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. *European Conference on Computer Vision*, pages 650–663, 2008.

[119] C Wolf, E Lombardi, O Celiktutan, and B Sankur. Real-Time Exact Graph Matching with Application in Human Action Recognition. *International Workshop on Human Behavior Understanding*, pages 17–28, 2012.

[120] W Xu, Z Miao, X P Zhang, and Y Tian. Learning a hierarchical spatio-temporal model for human activity recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1607–1611, March 2017.

[121] Y Yacoob and M J Black. Parameterized modeling and recognition of activites. *International Conference on Computer Vision*, pages 120–127, 1998.

[122] J Yamato, J Ohya, and K Ishii. Recognizing human action in time-sequential images using hidden markov model. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 379–385, 1992.

[123] L Zelnik-Manor and M Irani. Event-based analysis of video. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 123–130, 2001.

[124] D Zhang and D Gatica-Perez. Modeling individual and group actions in meetings with layered HMMs. *IEEE Transactions on Multimedia*, 8:509–520, 6 2006.

[125] Y Zhang, W Ge, M C Chang, and X Liu. Group context learning for event recognition. In *IEEE Workshop on Applications of Computer Vision*, pages 249–255, 2012.

[126] Y Zhao. Lecture 6: Diffusion distance. Lecture, Peking University, 2011.

[127] F Zheng, L Shao, and Z Song. Eigen-space learning using semi-supervised diffusion maps for human action recognition. *International Conference on Image and Video Retrieval*, pages 151–157, 2010.

[128] F Zheng, L Shao, Z Song, and X Chen. Action recognition using graph embedding and the co-occurrence matrices descriptor. *International Journal of Computer Mathematics*, 88:3896–3914, 6 2011.