# Video Sequence Alignment


**Manal Adnan Al Ghamdi**

Department of Computer Science

University of Sheffield

A dissertation submitted in partial fulfilment of
the requirements for the degree of

*Doctor of Philosophy*

Supervisor: Dr. Yoshihiko Gotoh

December 2014

In the name of God, most compassionate, most merciful

I would like to dedicate this thesis to my parents, who have been the biggest and continuous source of motivation throughout my life....

# Declaration

I hereby declare that this thesis is my own work and effort and that it has not been submitted anywhere for any award. Where other sources of information have been used, they have been acknowledged.

Manal Adnan Al Ghamdi
December 2014

# Acknowledgements

وَفَوْقَ كُلِّ ذِي عِلْمٍ عَلِيمٌ

"Over all those endowed with knowledge is the All-Knowing (ALLAH)".

In the name of Allah, the Most Gracious and the Most Merciful. Thanks to ALLAH who is the source of all the knowledge in this world, for the strengths and guidance in completing this thesis. It is a pleasure to attribute credit to the many people who have contributed directly or indirectly to this work.

I would firstly like to thank Dr. Yoshihiko Gotoh, for his supervision throughout my Masters and PhD studies. Without his guidance, motivation and support, this dissertation would not have been completed. One could not wish for more kind, accessible and friendlier supervisor. Special thanks to my panel members, Professor Guy J. Brown and Dr. Daniela M. Romano for their useful suggestions and being so kind to me. I am thankful to my examiners Jon Barker and Cathal Gurrin for useful discussion during the final viva of my defence.

I am hugely grateful to my parents for their support, prayers, love and care throughout my life; they have played a vital role in helping me to reach this milestone. Any success I have achieved during my personal life and my academic career I attribute to my brothers Khalied and Abdullah, to my sisters Mona and Malak, to my sister in-law Amani, relatives and friends for their continuous support and prayers.

I thank those who rendered their direct or indirect support during the period of my PhD work. The members of the Speech and Hearing group at Sheffield University have been, and continue to be, a source of knowledge, friendship and humour. My friends who never let me feel that I am away from my homeland: Sarah, Norah Alkhaldi, Ahmad, Usman, Nouf, Maryam, Samiah, Samah, Ohood, Hanan, Rabab, Norah Farooqi, Lie and Mohammed. I am also grateful to many individuals at the University of Sheffield who provided me with academic and technical support during my research.

Lastly, I gratefully acknowledge the financial support for my research from Umm AlQura University in Saudi Arabia, who gave me a scholarship to pursue my PhD studies.

# Abstract

The task of aligning multiple audio visual sequences with similar contents needs careful synchronisation in both spatial and temporal domains. It is a challenging task due to a broad range of contents variations, background clutter, occlusions, and other factors. This thesis is concerned with aligning video contents by characterising the spatial and temporal information embedded in the high-dimensional space. To that end a three-stage framework is developed, involving space-time representation of video clips with local linear coding, followed by their alignment in the manifold embedded space.

The first two stages present a video representation techniques based on local feature extraction and linear coding methods. Firstly, the scale invariant feature transform (SIFT) is extended to extract interest points not only from the spatial plane but also from the planes along the space-time axis. Locality constrained coding is then incorporated to project each descriptor into a local coordinate system produced by a pooling technique. Human action classification benchmarks are adopted to evaluate these two stages, comparing their performance against existing techniques. The results shows that space-time extension of SIFT with a linear coding scheme outperforms most of the state-of-the-art approaches on the action classification task owing to its ability to represent complex events in video sequences.

The final stage presents a manifold learning algorithm with spatio-temporal constraints to embed a video clip in a lower dimensional space while preserving the intrinsic geometry of the data. The similarities observed between frame sequences are captured by defining two types of correlation graphs: an intra-correlation graph within a single video sequence and an inter-correlation graph between two sequences. A video retrieval and ranking tasks are designed to evaluate the manifold learning stage. The experimental outcome shows that the approach outperforms the conventional techniques in defining similar video contents and capture the spatio-temporal correlations between them.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Measuring similarity between video sequences is one of the most important topics in computer vision and has been extensively studied recently; however, it is still in its infancy, especially in realistic and complex scenarios. The challenge mainly lies in the large variations, background clutter, occlusions, illumination changes and noise in these types of data. The task of visual analysis in video-processing comprises many aspects including representation, detection and tracking. In this thesis, we study the video sequences alignment problem and we specifically focus on video representation using different levels of features. We evaluate the work with applications in action classification, video search and retrieval and video ranking. In this chapter, we provide a brief background to the work, along with its motivation, scope and aim. We further define the main contributions of the work and its potential applications. Finally, the structure of the thesis is presented.

## 1.1 Background

Video search engines, such as Google Videos and YouTube, have become the most frequently visited websites. Search normally depends on annotations such as titles, tags and descriptions that are created manually. One potential problem is that annotations by the owner of a video do not always reflect the actual content of the video [Cheung and Zakhor (2003)]. Users tag their videos with all possible keywords to increase its spread throughout the web. Thus searching for popular video keywords such as 'Star Wars' or 'Clinton testimony' returns a tremendous number of similar video clips. In addition, manual annotation to make online data more accessible is a comprehensive and tedious process that requires human intervention. Another problem is that there

is no central management to protect copyright of these data, so duplication of content is highly possible [Shivakumar and Garcia-Molina (1999)]. Similar versions of the same video can be found in multiple locations with different names and formats to facilitate downloading and streaming.

## 1.2 Motivation

Identifying similar content can benefit various applications and content owners. For example, it will reduce the number of retrieved data through a web search and identify inappropriate use of copyright content. Many algorithms have been proposed to represent, search and retrieve visual content in multimedia data. However, it is still a big challenge to access video content efficiently and help users gain useful information relevant to their interests. Three types of information have been utilised to represent video sequences: visual information contained in the frames content, audio information from the audio channel, and text information, which consists of tagged texts embedded in the video containing title, summary, date, duration, file size, video format and so on [Hu et al. (2011)]. Some works use a fusion of these different types of information. Others convert audio content into text information using automatic speech recognition (ASR) engines and solve the problem with text-processing techniques. However, most of the state-of-the-art results in video-processing applications are obtained by considering visual content only [Feng et al. (2003)]. Dealing with visual content is more accurate for video-processing than depending on video metadata created by the video's owner which may not reflect the actual contents, or on the audio signal that may contain more than speech, such as in home videos. In addition, considering a single type of information saves both processing time and storage space for video-processing applications.

Aligning audio visual sequences in both spatial and temporal domains is fundamental for various applications including that of video similarity [Caspi and Irani (2002)]. It solves the spatial ambiguities, deals with cases that can not be covered with image-to-image alignment approaches and allows more advanced applications where image-to-image alignment fails. Image-to-image alignment refers to the problem of estimating correspondences between two or more images, *i.e.* finding for each pixel in one image its corresponding pixel in the other image. Video sequences capture much more information than any individual frame does. Scene dynamics (e.g. object movement), non-rigid changes in the scene (e.g. flowing water) and changes

in illumination are examples of information found between the frames that cannot be captured by representing individual frames.

Representing a video sequence is the key step for alignment techniques. In video data a frame sequence is chronologically tied to describe a story. Each frame is sampled to capture a moment, forming a coherent relation with the adjacent frames. Their spatial and temporal coherence is embedded in the high-dimensional space, building a complex structure of video. The spatio-temporal correlation can be derived from frame sequences to represent similarity across frame instances. A manifold is then learnt for these sequences in order to map a high-dimensional video representation into a lower-dimensional space. A series of intrinsic coordinates for each manifold is generated in chronological order by context of spatial and temporal similarity. As a consequence the identification of corresponding frames across video sequences becomes a simpler task. Such representation can provide a basis for various applications including video searching and retrieval, human activity understanding, video summarisation and copy detection. It can also be extended to represent various types of video content for different applications such as sports video analysis and video surveillance.

## 1.3 Scope and Aim

There are two main questions about applications that involve finding a relationship between two video sequences. The first is how to interpret the video events and the transition between them. The second is how to define the relationship between two videos using the outcome of the first question. This thesis focuses on these two areas related to video representation and measuring the similarity between multiple videos, and to that end considers a task of aligning similar contents in video streams with variations in motion performance, recording settings and inter-personal differences. The problem is important as it is a heart and core of many video-processing applications.

To answer the first question, a three-stage scheme is developed involving space-time representation of video clips; their alignment in the manifold embedded space relates to the second question. Stage 1 extends one of the popular feature extraction approaches in the image-processing domain to consider the temporal relationship between the features. The interest points that have significant local variations in both space and time are extracted, and the regions around them are described as video representation. In stage 2, using a features-learning technique, a dictionary of the extracted features is created to convert the low-level features defined from the pixels of the frames to

mid-level features containing a set of codes with richer representation. In stage 3, a manifold learning algorithm with spatio-temporal constraints is defined to embed a video clip in a lower-dimensional space while preserving the intrinsic geometry of the data. The similarities observed between frame sequences are captured by defining two types of correlation: an intra-correlation within a single video sequence and an inter-correlation between two sequences. Finally, different applications related to the video similarity task are developed to evaluate the effectiveness of the developed techniques.

## 1.4 Thesis Contributions

This section lists the thesis contributions, describing the background behind the motivation and need for the work carried out by the thesis.

### Contribution 1: Video sequence alignment framework

**Background.** Many computer vision applications involve processing multiple videos recorded simultaneously from different angles. In these applications, alignment and synchronisation are essential if a consistent structure from the videos is to be recovered. 'Alignment' involves finding the temporal correspondences between two video sequences, which cannot be captured by standard image-to-image alignment techniques [Caspi and Irani (2002)]. Image-to-image alignment methods deal only with spatial variation for scene appearance (such as object characteristics) contained in the individual images. However, there are some cases where the common spatial information between two images is not enough for reliable alignment. Video sequences contain temporal variation for dynamic scenes (such as moving objects or changes in scene illumination) that cannot be captured by matching two images and can only be found in the transition from one image to another.

**Contribution.** This thesis develops a three-stage framework to align contents in video sequences using spatio-temporal representation, coding techniques and manifold embedding. The developed work is novel in that it does not utilise any prior information, such as the length and content of the scenes. Unlike the previous works, the developed framework does not required template matching or pre-processing steps. Instead it analyses a video by characterising the spatial and temporal information embedded in a frame sequence. A video is represented by spatio-temporal features, which are analysed in the lower-dimensional space. Evaluation of different tasks showed that

the developed framework attained a significant improvement over the conventional alternatives in estimating the relevance between videos . Multiple types of datasets with different characteristics are utilised in the development, and the evaluation processes proved the framework capable of dealing with variations in scales, viewing angles, and background conditions (both indoors and outdoors).

## Contribution 2: Space-time video representation

**Background.** Recently, spatio-temporal local features have been introduced into video-processing applications. They share some of the same properties as the spatial features of images (such as accuracy, stability, scale and rotation invariant) and they achieve good performance in video-processing tasks [Chen and Hauptmann (2009)]. The straightforward way to extract a spatio-temporal interest point is to extend a two-dimensional (2D) spatial interest point detection method to the temporal domain. Some recent work has utilised motion techniques, such as optical flow or a Gabor filter, to capture the temporal information [Dollar et al. (2005); Laptev and Lindeberg (2003)]. However, these algorithms suffer from sensitivity to different changes, their extracted points are restricted to detecting a single action in the scene, and they are not able to capture the geometric structure of the action parts [Shechtman and Irani (2007)].

**Contribution.** The first contribution is the development of the ST-SIFT detector to extract interest points that have significant local variations in both spatial and temporal domains and are invariant to different spatio-temporal variations such as scale, location and orientation. Since its original development by Lowe (2004), scale invariant feature transform (SIFT) has been successfully applied to various image-processing studies for locally detecting and describing interest points. It has proved its efficiency with tasks in a 2D space, for example with image segmentation and classification. No works have been reported however to fully extend this approach in the spatio-temporal domain. Such an extension can play a significant role in different video-processing applications.

## Contribution 3: Spatio-temporal Coding

**Background.** Feature-learning and coding methods have been widely used in image-processing applications to obtain global representations with fewer codes than the extracted features. However, these methods disregard the information about the spatial

relationship of the features; hence they are incapable of capturing shapes or locating an object [Wang et al. (2010)]. Many extensions to the conventional method originally developed for the image-processing domain have been introduced by adding the locality constraints to project the descriptors into their local-coordinate system. The projected coordinates are then integrated by a pooling technique to generate the final representation. They have been efficiently used for various image applications, such as classification and recognition. These extensions may consider the information about the spatial layout of features that help to capture shapes or locate an object; however, there is still a gap in considering the temporal information at the coding stage.

**Contribution.** The second contribution adopted one of the most effective techniques in the sparse coding area from the image domain to the video domain. The sparse representations have been efficiently used in image-processing to encode a set of features using only a few active coefficients; this makes the encoding easy to interpret and reduces the computational cost. The coding technique represents video content with fewer codes than the original set of low-level features, which helps to save both processing time and storage space for the visual descriptors.

## Contribution 4: Spatio-temporal Manifold Representation

**Background.** The task of aligning multiple audio visual sequences from different angles needs careful synchronisation in both spatial and temporal domains. The majority of previous work in this area employed techniques such as template matching, camera calibration analysis and object tracking. A number of techniques have been proposed to discover the underlying structure of high-dimensional data, such as PCA (Principle Component Analysis) [Wold et al. (1987)] and Isomap (Isometric Feature Mapping) [Tenenbaum et al. (2000)]. These methods work well with curve-shaped data or non-linear structures. However, none of them cover the temporal relationship between the points, which is required in video-processing applications.

**Contribution.** The final contrition is a manifold embedding technique with spatio-temporal graph between frames to synchronise and align video content in lower-dimensional space. It defines the relationship between a pair of video sequences by reordering and clustering their frames into groups based on spatio-temporal similarity.

## 1.5 Justification For the Research

Typical applications such as search engines, videos on demand and distance learning involve searching and retrieving huge amounts of video data [Chen and Chua (2001)]. The usefulness of these applications depends on whether the retrieved videos match user interest. Because of the wide-varying interpretation of visual content, it is not effective to depend only on textual information to represent video data. Search techniques employ visual and audio information as an intuitive alternative. However, audio and visual integration-based approaches are limited because it is difficult to define the relation between audio and visual segments.

An alignment process is often necessary for applications that require users to find the temporal relation between a pair of video sequences. Defining the video trajectory in a lower-dimensional space helps to highlight the important transactions in the scene and the common one between two scenes [Caspi and Irani (2002)]. Considering these trajectories is an effective tool for solving video problems. It can be applied to natural imagery of non-rigid motions and does not require prior information about object characteristics or the duration of the video. In addition, defining video trajectory does not involve pre-processing steps for the video data, such as background subtraction, object detection and tracking, template matching, or segmentation. This makes it computationally less expensive, needing less storage capacity and delivering faster processing times.

### 1.5.1 Application Areas

Various approaches have been introduced to tackle the problem of discovering correspondence between video sequences. These can be beneficial to content owners and video applications such as:

- **Video searching and retrieval:** With the explosion of electronic devices (such as tablets and smart phones), internet usage, and online publishing, users have easy access to a massive number of videos. However, their chances of effectively indexing and manipulating such a huge quantity of data are rather limited. Currently, web search engines (such as Google, Yahoo and Bing) rely mainly on natural language descriptions to retrieve the required videos. Their performance depends on the keywords used to tag these videos rather than the actual visual content. Video alignment can provide a searching tool that retrieves the video

of interest based on video queries, which also involves locating a short clip in a video stream.

- **Human action recognition:** Detecting and recognising human activities in video sequences is a challenging task, due to videos' variations in motion, viewpoint, illumination, camera angles and lighting effects. A set of labels is usually predefined for some action classes. For each class positive and negative training samples are given, from which a classifier is constructed. Given a test video sequence, the purpose is to identify the corresponding action class label. Therefore, the task is considered to be finding the correspondences between a test and training video, where the video alignment approaches become useful.

- **Video copy detection:** This task is part of the near-duplicate detection task that has attracted research attention recently due to the exponential growth of social media on the internet and the huge volume of videos being transmitted and searched. Near-duplicate videos are either identical or approximately identical to the original versions. They may have similar or different duration, file formats, encoding parameters, photometric variations (colour, lighting changes), editing operations (caption, logo), and certain modifications (frames add/remove). Detecting these copies is a challenging task because of the multiple geometric and photometric transformations caused by different camera angles, digitalisation and editing steps. Defining correspondence between these types of videos is a challenging task and needs careful synchronisation in both spatial and temporal domains. The alignment approach can help to estimate correspondences between two video sequences.

- **Video mining and summarisation:** Summarisation applications usually help to save storage space without losing the descriptive content of the video, and provide efficient and quick retrieval results for users. The main step in this application is to define the crucial content that represents the main events in the video sequence. Removing similar or repeated parts and defining the action performed can be key to summarising video content. From this point of view, representing the video events and detecting repetitions become useful for the summarisation task.

## 1.6 Thesis Overview

Figure 1.1 represents the structure of the thesis. The boxes show the numbers of chapter or section of the associated chapter with their titles. The black arrows shows the dependency between the chapters. For example, Chapter 6 builds on the results from Chapter 3, 4 and 5. Red arrows indicate a lesser degree of dependency. For example, each one of Chapter 4, 5 and 6 presents a brief overview of the previous chapters as techniques used in its experiment therefore, it is possible to read Chapter 5 for example before Chapter 4 and 3, or Chapter 4 before Chapter 3. Blue arrows indicate the tasks used to evaluate the chapter with the associated datasets. For example, Chapter 3 and 4 used the human action recognition task for evaluation.

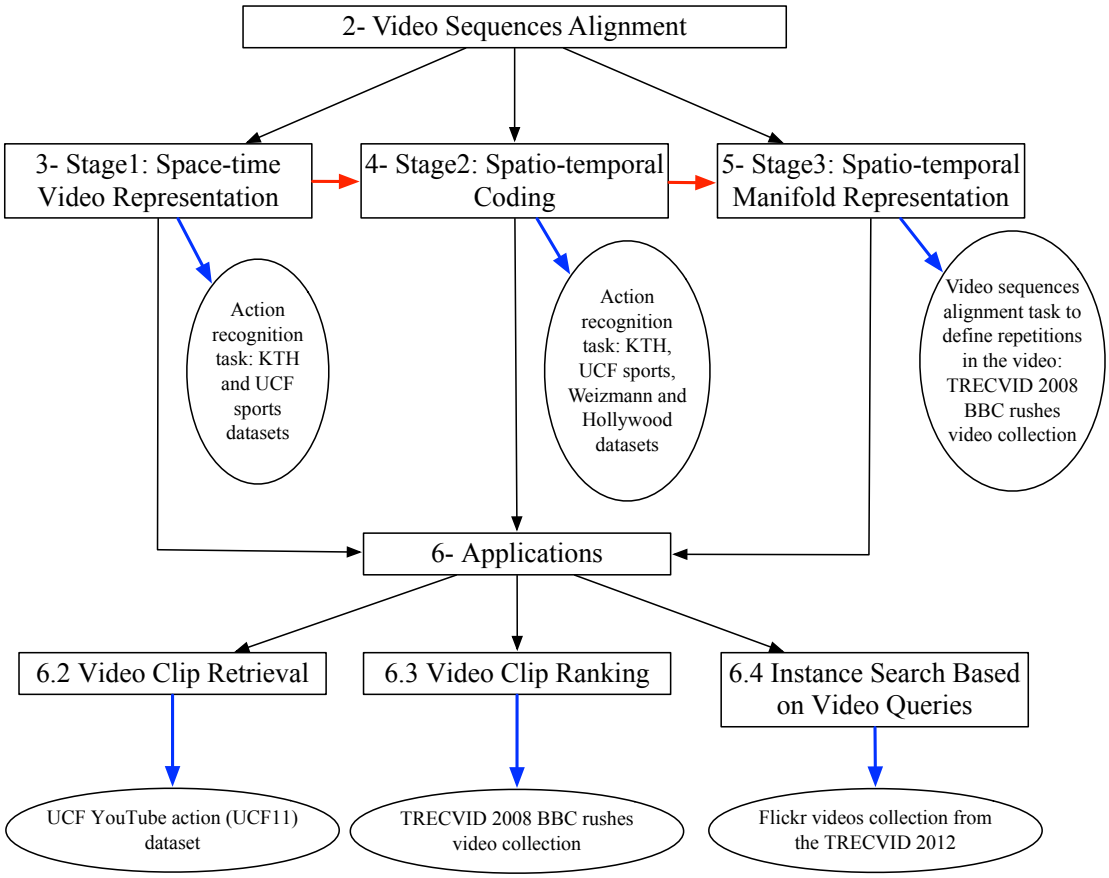



Figure 1.1: Thesis structure. Each box represents a chapter or section of the associated chapter. A black arrow indicates the dependency between the chapters. A blue arrow indicates the defined task and the datasets used for the associated task.

Chapter 2 provides a general overview of the framework developed in this thesis as

well as the datasets used to evaluate each technique and the final applications defined to evaluate the entire framework. Chapter 3, 4 and 5 contain the main contributions of this thesis, while Chapter 6 contains three applications from the video similarity domain used to evaluate the framework developed through out the thesis. An overview of each chapter is summarised as follows.

- **Chapter 2: Video Sequences Alignment Framework**
  This chapter presents a framework for video representation and alignment with three stages each representing a different type of features. A breakdown of this framework with more detail is presented in the following chapters. This chapter justifies contribution 1.

- **Chapter 3: Stage 1: Spatio-temporal SIFT**
  This chapter introduces the first stage of the developed framework, which involves a space-time extension of the SIFT originally applied to the 2D volumetric images. It also presents the experiment results on the human action classification task. The contents of this chapter are based on the paper [Al Ghamdi et al. (2012b)] and justify contribution 2.

- **Chapter 4: Stage 2: Spatio-temporal Coding**
  This chapter presents a spatio-temporal coding technique for a video sequence as second stage. It is based on the feature extraction technique combined with the coding technique. It also demonstrates experiments on the human action classification task. This chapter is based on the paper [Al Ghamdi et al. (2012a)] and it justifies contribution 3.

- **Chapter 5: Stage 3: Spatio-temporal Isomap**
  The last stage is presented in this chapter including a spatio-temporal graph-based manifold embedding to aligning video contents in a lower-dimensional space. Experiment results to identify the repetitive contents from complex scenes are also presented. This chapter is based on the papers [Al Ghamdi and Gotoh (2013, 2014a)] and it justifies contribution 4.

- **Chapter 6: Applications**
  This chapter provides some applications whose results can be used to evaluate the developed framework. This chapter is based on the papers [Al Ghamdi and Gotoh (2014b,c)].

- **Chapter 7: Conclusions and Future Work**
  The last chapter concludes this research by providing a summary, recommendations and suggestions for the direction of future work.

## 1.7 Published Work

This thesis is partly based on the following publications.

1. Spatio-temporal SIFT and Its Application to Human Action Classification. *Manal Al Ghamdi, Lei Zhang, and Yoshihiko Gotoh*, In *12th European Conference on Computer Vision (ECCV), VECTAR workshop, 2012*

2. Spatio-temporal Video Representation with Locality-Constrained Linear Coding. *Manal Al Ghamdi, Nouf Al Harbi, and Yoshihiko Gotoh*, In *12th European Conference on Computer Vision (ECCV), ARTEMIS workshop, 2012.*

3. The University of Sheffield and Harbin Engineering University at TREC video 2012: Instance Search. *Manal Al Ghamdi, Muhammad Usman Ghani Khan, Lei Zhang and Yoshihiko Gotoh*, In *TREC Video Retrieval Evaluation, 2013*

4. Spatio-temporal Manifold Embedding for Nearly-repetitive Contents in a Video Stream. *Manal Al Ghamdi and Yoshihiko Gotoh*, In *The 15th International Conference on Computer Analysis of Images and Patterns (CAIP), 2013*

5. Alignment of Nearly-duplicate Contents in Video Stream with Manifold Embedding. *Manal Al Ghamdi and Yoshihiko Gotoh*, In *The 39th International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2014*

6. Video Clip Retrieval by Graph Matching. *Manal Al Ghamdi and Yoshihiko Gotoh*, In *36TH European Conference on Information Retrieval (ECIR), 2014*

7. Manifold Matching with Application to Instance Search based on Video Queries. *Manal Al Ghamdi and Yoshihiko Gotoh*, In *The 6th International Conference on Image and Signal Processing (ICISP), 2014*

8. Video Sequences Alignment. *Manal Al Ghamdi and Yoshihiko Gotoh (Submitted for Image and Vision Computing, 2015)*

# Chapter 2

# Video Sequences Alignment Framework

The focus of this thesis is to define and develop a three-stage video sequences representation and alignment framework, and then validate this framework using different video similarity applications. It is crucial to understand the purpose of each stage individually and then the relationships between them.

The purpose of this chapter is to introduce the video sequences alignment framework and some details of the three stages and how each contributes to solving the problem of video representation and similarity measurement. It also reviews some work relating to video sequences alignment applications. Additionally, it describes information which underpins the work presented in the following chapters, including the datasets used throughout the work and the applications developed to evaluate the framework performance. More technical details for each stage are presented in the following three chapters.

## 2.1 Introduction

Video sequences alignment is often required for computer vision applications that involve processing of multiple videos. Alignment involves defining the temporal relationship between two video sequences based on their feature trajectory. A feature trajectory is the sequence of points, either static or dynamic, that define the feature's location in each frame along the video [Caspi and Irani (2002)]. Spatio-temporal alignment between a pair of video sequences can then be defined by establishing correspondences between their trajectories. Feature-based image alignment methods have

been used in image processing to find correspondences between a pair of images [Torr and Zisserman (2000)]. They work by extracting local interest points from each image and then defining the corresponding points using either robust estimation methods such as RANSAC [Fischler and Bolles (1981)] or LMS [Hampel et al. (2011)], which extract spatial transformation and estimate corresponding points between the two images, or by using a correlation-based matching method [Xu and Zhang (1996)] that initialises the approximation of matching features based on the brightness values of the point's neighbourhoods.

The problem of image-to-image alignment focuses on finding corresponding points between two or more images, *i.e.* finding for each feature or interest point in the first image a corresponding feature or interest point in the second image. In this study the feature-based image-to-image alignment was generalised to feature-based trajectory-to-trajectory alignment by considering the feature points as feature trajectories. This means estimating corresponding features between two videos based on their spatio-temporal relationship, *i.e.* finding for each feature at a specific frame in the first sequence its corresponding time and position in the other sequence.

In this thesis, an analysis of the video content was carried out by characterising the spatial and temporal information embedded in a frame sequence. Then the problem of measuring the similarity between video sequences was formulated as finding the correspondences between their feature trajectories in the lower-dimensional space. Firstly, to define the feature trajectories, the video events were represented with interest points that have significant local variations in both space and time. These features were then encoded with fewer codebook bases than the original number of interest points, to define a more compact representation. Finally, these codes were projected into the lower-dimensional space to discover the underlying structure in the sequences. Many problems can be solved by analysing the generated coordinates in the lower-dimensional space, which are chronologically ordered by the spatio-temporal similarity and connected as trajectories to be used for sequences alignment.

The following sections review recent works developed to solve alignment and similarity of video sequences, and provide some details about the developed approaches, along with the datasets used for development and the applications used for evaluation.

## 2.2 Related Works

This section provides an overview of important approaches regarding alignment and similarity of video sequences. Additionally, nearly-repetitive content detection is also reviewed, since this is a special task of the video copy detection and video similarity domain, and is related to one of the developed applications in Chapter 6.

### 2.2.1 Video Sequences Alignment

Image-to-image alignment refers to defining the matching or similar points between two or more images based on their spatial relationship. Unlike images alignment, where the core problem and the related applications have been extensively studied (such as [Bergen et al. (1992)],[Zhang et al. (1995)],[Zoghlami et al. (1997)]), only a few works have addressed the video sequences alignment problem. In the literature, there are two classes of sequences-aligning algorithms: sequence-to-sequence (or direct approach) and trajectory-to-trajectory (or features-based approach) [Rao et al. (2003); Wedge et al. (2005)]. The sequence-to-sequence approach applies the computation over all pixels in the video frames. The trajectory-to-trajectory approach uses the feature points to track the movement in view, and applies the computation to the defined trajectories. The merit of the direct approach is that it does not include a feature detection and tracking step, and it represents the spatial transformation between sequences more accurately than the trajectory-to-trajectory approach. On the other hand, the trajectory-to-trajectory approach considers geometric information, and is therefore more able than the direct approach to align video sequences acquired by various sensors, and is less affected by background changes.

Many trajectories-based works have been proposed based on a geometric entity such as the fundamental matrix or homography, which involves invertible mapping of points from one plane to another [Rao et al. (2003)]. When one of these geometric entities or characteristics is estimated from the stationary background of a scene, the alignment can be established by calculating the re-projection errors of moving points. For instance, Reid and Zisserman (1996) aligned two videos of a soccer match by firstly utilising the ground markings to relate the homography of the ground plane between two scenes. Then, by minimising the re-projection errors of players' shadows on the ground, the two videos can be aligned. Similarly, Stein (1999) proposed a method of aligning tracking points obtained from different cameras, assuming a homographic relationship between the cameras. Stein performed an exhaustive search of the dif-

ferent intervals between video sequences to discover the temporal alignment. This method was computationally expensive, and it required a constant time shift to align the video.

In video surveillance, Caspi and Irani (2002) aligned two videos by finding the spatio-temporal transformation that minimises the sum of squared differences (SSD) between the sequences. For action recognition, Giese and Poggio (1999) utilised the dynamic shift of the time stamp of the spatial information to achiev the spatio-temporal alignment between two video sequences. They represented the 2D action trajectory by a linear combination of prototypical views, and expressed viewpoint changes by varying the coefficients of this linear combination. Their method can only align simple motion patterns, because they did not use 3D information.

### 2.2.2 Video Sequences Similarity

Video similarity measurement has played an important role in various video-processing applications. Depending on the query type, methods to measure video similarities can be categorised into three types; feature-matching, text matching, or ontology-based matching [Hu et al. (2011)].

**Feature-Matching:** The straightforward method to measure the similarity between a pair of video clips is to calculate the average distance between the features of the corresponding frames [Hu et al. (2011)]. Usually, low-level features are used to define relevant videos. However, depending on the user's interest and the problem to be solved, different level of features can be employed for the similarity measurement, such as static features of key frames, object features or motion features. For example, Sivic et al. (2005) matched extracted face features from an example shot containing a queried face with stored face features and then those with the queried face were retrieved. Lie and Hsiao (2002) represented the major objects in a set of videos with trajectory features and then matched them with the stored trajectory features to retrieve similar videos. The benefit of feature-matching methods is that the video similarity can be directly measured in the feature space. However, it is hard to represent the semantic similarity because of the gap between features vectors and the humans perception of the semantic categories.

**Text matching:** This is the simplest method of finding similar videos for a given query: by matching the name of concepts with query terms [Hu et al. (2011)]. For example, Hauptmann et al. (2007) solved this by firstly normalising both the descrip-

tions of concepts and the query text, then computing the similarity between them by applying a vector space model, and finally selecting the concepts with the highest similarity score. These approaches have the advantage of intuitiveness and simplicity of implementation [Hu et al. (2011)]. However, in order to obtain good search results, they require all concepts to be explicitly included in the query text.

**Ontology-based matching:** This method depends on the defined ontology between semantic concepts or semantic relations between keywords [Hu et al. (2011)]. Query descriptions are combined with extra information from knowledge sources, such as ontology of concepts or keywords. Hauptmann et al. (2007) translated the nouns and noun chunks defined in the text query to ontological concepts using Wordnet. By linking the concepts to Wordnet, they used the ontology to find the concept most related to the original query text. Based on the fact that visual co-occurrence can be approximated by semantic word similarity, Yusuf Aytar (2008) defined similar videos to a particular query by measuring the similarity between videos' text annotation and the user-defined text query. These methods are good for using extra concepts from knowledge sources that improve retrieval results [Hu et al. (2011)]. However, irrelevant concepts might also be involved, resulting in unexpected deterioration of search results.

Detecting repetitive content in video sequences can be considered a special case of the video similarity problem. The task can be formulated as finding similar, but not identical, content within a single video sequence. Several works consider the analysis of video streams with repetitive content (such as the rushes[1]). Summarisation, retrieval and indexing are the most popular tasks for this kind of development data.

Various publications related to the summarisation task have been published as part of the annual TRECVID competition [Over et al. (2008)]. Bailer et al. (2007), for example, developed a skimming algorithm for the rushes video. They firstly remove the unusable content such as the colour bars and black frames, eliminate redundancy by clustering the multiple retakes of the same scene and finally select the representative one for the scene. Similarly, Ren et al. (2008) proposed a system for detecting redundancy in the rushes collection. They cluster related shots based on the similarity between the extracted key frame from each shot. Similar shots from the same cluster are removed due to repetition and the shot of the centre key frame is used in the video summary.

For rushes parsing, Dumont and Merialdo (2009) presented a method for detecting

[1]Rushes is a raw material which will be further used to evaluate the developed framework

repetitive segments and then parsing the video content into scene and retakes. The video content is subsequently decomposed into one-second segments and a hierarchical clustering from these segments is defined. The multiple retakes for the same scene are then defined and the most representative one is selected.

Another approach by Benini et al. (2009) maps the rushes content into a geometrical trajectory in a multi-modal space for similarity and retrieval tasks. The video content is characterised by three axes called natural $N$ (for the shape), temporal $T$ (for the movement) and energetic $E$ (for the behaviour) dimensions. After defining these features, each point of the trajectory can be represented in the multi-modal space by a triple value $(N, T, E)$. Then, to reduce the complexity of comparing two trajectories, a 3D-solid shape $S$ is defined as a representation of the fundamental characteristics of the video. The similarity between two different rushes is defined by a multi-modal distance $D$ between their solids $S_A$ and $S_B$.

In conclusion, most of the approaches divide the video stream into shots, clips or segments and then cluster these units into related groups. From each cluster, a representative is defined to represent the unit and a summary of the video is generated.

## 2.3 Framework Description

Measuring similarity between video sequences is at the heart of most computer vision applications. Trajectory-to-trajectory alignment is one method inspired by the success of image-to-image matching methods in spatial domain applications. Most previous works have focused on applying sliding windows and template matching to align video sequences. However, these methods are computationally expensive in accurately localising actions, and they have difficulty in detecting actions that mainly occur in the space-time domain, such as flowing water, because they cannot be spatially segmented [Aggarwal and Ryoo (2011)]. Therefore there is a need to firstly define a useful video representation and then a similarity measurement that can be a part of different video-processing applications. In this thesis, a three-stage framework is proposed to represent and align video content in lower-dimensional space. The architecture of this framework is shown in Figure 2.1.

In the first stage of the presented framework, the content of the video stream is described in the high-dimensional space using a spatio-temporal extension of one of the 2D interest point detectors that demonstrated great success in different image-processing applications. Defining a video representation is the core step in analysing

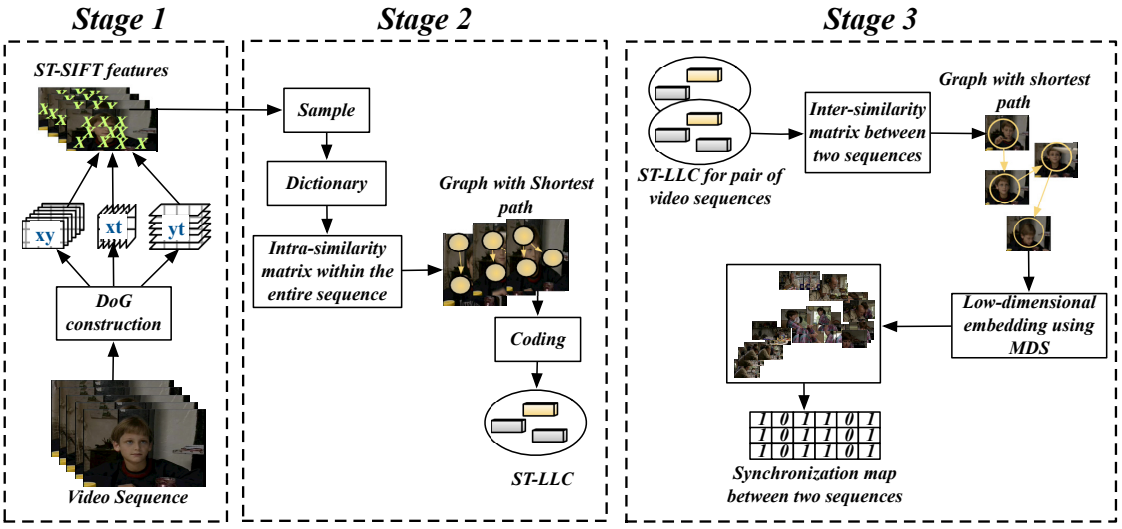



Figure 2.1: The developed framework for video sequences alignment. There are three major stages: spatio-temporal local feature extraction, video coding construction, manifold learning with spatio-temporal graph. The input video volume will be mapped to a trajectory of coordinates in the lower-dimensional space that is evaluated later with three different applications in video searching and retrieval task.

video content. Various approaches have been presented to solve this problem; however, developments are still in their infancy. Some of these approaches capture appearance and motion separately [as Chen and Hauptmann (2009)], which is computationally expensive, and they may achieve invariance in the spatial domain but not in the temporal domain, since they treat it separately. In addition, these methods may fail to detect long-period actions (such as running) because they consider temporal motion features within a few frames. Popular spatio-temporal interest point detectors [such as Dollar et al. (2005), Laptev and Lindeberg (2003)] extended established 2D interest point detectors, such as the Harris corner detector [Harris and Stephens (1988)], to the spatio-temporal domain. The number of defined interest-points in these methods is small and often not enough to characterise complex sequences. In addition, the original 2D approaches have many limitations in the spatial domain, including sensitivity to the various variations, which may be inherited by their extensions. Furthermore, they mostly apply pre-processing steps such as background subtraction or post-processing steps such as filtering. To that end, identifying spatially distinctive interest points that exhibit sufficient motion at a variety of spatial scales is a requirement for video representation.

The second stage generates a mid-level representation from the extracted features

using a coding technique which considers the locality of the manifold structure in the input space. Bag-of-words (BoW) methods have been widely used in the literature for different tasks as local spatio-temporal descriptors [such as Chen and Hauptmann (2009), Laptev (2005), Klaser et al. (2008), Laptev et al. (2008)]. However, the BoW methods have an important limitation: they largely ignore the spatio-temporal relationship between the local features, such as temporal order of the action, spatial arrangement of objects, and motion trajectories [Wang et al. (2011)]. This second stage addresses this problem by representing the spatio-temporal descriptors by a set of linear codes based on a spatio-temporal graph.

The third and last stage is to embed the generated representation in the lower-dimensional space and define a video trajectory of coordinates that can be used later for different applications. At this stage the similarity is computed between video contents using the spatio-temporal graph.

The various traditional approaches from the image-processing domain were extended to the video (3D spatio-temporal volume) domain. The developing approaches are evaluated using three applications within the video searching and retrieval task. Details of each stage implementation are presented in the following chapters. However, the general idea underpinning each stage are given below.

## Stage 1: Space-time Video Representation

In order to analyse the video content, the volume of pixels needs to be reduced into a descriptive set of features. Alignment tasks need a robust representation which is invariant to various changes and is able to describe complex and realistic events in a spatio-temporal domain. This is hard to achieve with the current shortcomings of existing methods (discussed in Section 2.2) such as sensitivity to variations, applying pre- and post-processing steps, using a sliding window, and ignoring the relationship between the spatial and temporal information.

To solve this problem, the first stage presents ST-SIFT as a space-time extension to the SIFT originally applied to the 2D volumetric images [Lowe (2004)]. SIFT has proved to be an effective method in image-processing, extracting interest points which are invariant to translation, scale, rotation, affine transforms, change in illumination, change in 3-D viewpoint, *etc.*In addition, it allows for correct object detection with low probability of mismatching and its points are easy to match against a huge set of local features [Zeng et al. (2011)]. This is because the defined points are highly distinctive and relatively easy and straightforward to extract. Extending these characteristics to

both spatial and temporal domains is beneficial for various video-processing applications.

A spatio-temporal difference-of-Gaussian (DoG) pyramid is firstly constructed to detect the local extrema, aiming at processing video streams. Interest points are then extracted from the spatial plane ($xy$) as well as the two planes along the time axis ($xt$ and $yt$), and the regions around them are described as video representation (as shown in the left panel of Figure 2.1). The ST-SIFT was evaluated using the human action classification task.

### Stage 2: Coding Technique

Feature extraction methods generate a tremendous number of descriptors, especially in representing video sequences, and the spatio-temporal nature of video sequences adds more complexity. Modelling spatio-temporal dependencies for 3D video data is a challenging task [Wang et al. (2011)]. To simplify it, previous works have usually applied schemes such as the BoW method, which assumes conditional independence across the spatial domain and represents video content by a histogram of feature occurrences. Various BoW approaches using local spatio-temporal descriptors have been proposed in the literature. However, they failed to consider the spatio-temporal relationship between the local features.

For that, the second stage derives spatio-temporal codes as a mid-level representation of the video stream given the ST-SIFT descriptors. The LLC is a coding scheme proposed by [Wang et al. (2010)], which is extended at this stage to the space-time domain to encode individual descriptors with their respective local coordinate systems. LLC performed remarkably better than the other coding techniques in different image-processing tasks. It is also considered simple, fast, scalable, and competitive. At this stage, intra-correlation within each video sequence (between frames) is derived by firstly constructing a shortest path graph between the descriptor set and a learned codebook, performing a k-nearest neighbour (kNN) search, and finally by solving a constrained least-squares fitting problem. The above procedure is outlined in the middle panel of Figure 2.1. Similar to Stage 1, the algorithm was evaluated using the human action classification task.

## Stage 3: Manifold Learning

Dimensionality reduction has been introduced to solve the problems in high-dimensional space; examples are eliminating the noise features in the data set and reducing the number of features [Law (2006)]. This can also reduce the computation time and space needed to store the features. Models with a low number of variables are often easier for the domain experts to interpret.

Dimensionality reduction is also used as a visualisation tool that transforms the high-dimensional into two- or three-dimensions for display purposes, which can provide an additional insight and helps to solve many problems. The main drawback of dimensionality reduction methods is the possibility of information loss, especially in the spatio-temporal domain. They ignore the information from the temporal dimension found in a multimedia sequence; hence they are unable to address the similarity-mapping problem in the spatio-temporal domain.

The final stage of the alignment framework addresses this problem by extending the spatial Isomap [Tenenbaum et al. (2000)] to spatio-temporal graph-based manifold embedding (or STG-Isomap) in order to capture the correlations between two sequences. It maps the high-dimensional representation to a spatio-temporal manifold representation where nodes represent frames and edges represent the temporal order (event sequence).

Isomap is the primary focus for spatio-temporal embedding in this thesis, because it can be extended as a result of its dependence on a specific distance measurement between data points (due to its foundation in multidimensional scaling) [Jenkins and Matarić (2004)]. Additionally, Isomap is much more powerful compared with other manifold embedding and can deal with a greater number of manifolds without any loss [Lee and Verleysen (2007)]. At this stage, the inter-correlation is defined between two sequences by constructing a shortest-path graph with a kNN search in both the spatial and the temporal domains.

The developed method reconstructs the frame's order based on their spatio-temporal relationship and recalculates distances along the sequence to ensure the shortest distance. A manifold is then learnt for these sequences to map high-dimensional video codes into a lower-dimensional space, as shown in the right panel of Figure 2.1. This approach was evaluated using a nearly-repetitive content detection task in the video stream.

## 2.4 Datasets

The developed techniques in this thesis were evaluated on different datasets: the KTH dataset [Schuldt et al. (2004)], the Weizmann dataset [Blank et al. (2005)], the UCF sports dataset [Rodriguez et al. (2008)], UCF YouTube Action (UCF11) [Liu et al. (2009b)], Hollywood dataset [Laptev et al. (2008)], the rushes video dataset [Over et al. (2008)] and the Flickr videos dataset [Over et al. (2012)]. The evaluation started with simple and standard datasets (KTH and Weizmann), used in most human activity analysis researches, and then moved to more challenging and realistic benchmarks (Hollowed, UCF sports, UCF YouTube and Flickr), with a wide range of scenes and viewpoints in complex environments. Finally we raise the challenge further and used the rushes video collection, consisting of a large amount of raw material with highly redundant contents.

Various datasets are available for video processing applications. Choosing between them was based on the task defined for evaluation. In the human action recognition task, the datasets used were firstly standard that give us the chance to compare our results with the state-of-the-arts. Additionally, they contains different variations and and viewpoints changes to prove the techniques strength and to evaluate it is ability at different situation. All these datasets were the most popular in the domain and have been used by most of the researches, other datasets were newly developed and fewer works employed them for the potential tasks.

Chapter 6 contains three applications that required a special case datasets. One task was the Instance search task the locate recognizable objects defined by video query in a collection of video clips. The original task defined for the TRECVID competition created a special type of dataset from the Flickr website. Therefore, generating a subset from the original dataset was the most suitable one. Chapter 5 evaluates the developed technique using by detecting the nearly repetitive contents in video stream. The rushes video collection was the only dataset in the video-processing domain with repetitive contents, which makes it the most suitable one for the defined task. A summary of these datasets are given in Table 2.1.

### KTH

The KTH dataset was created by Schuldt et al. (2004) as a collection of human actions with six different activities: walking, jogging, running, hand-waving, boxing and hand-clapping. Each action is performed by 25 people in four different scenarios outdoors

| Dataset | Classes | No. of Videos | Resolution | Views | Properties |
|---|---|---|---|---|---|
| KTH | 6 classes: walking, jogging, running, boxing, hand waving, and hand clapping | 600 | $160 \times 120$ | frontal/side | Homogeneous indoor /outdoor backgrounds, performed by 25 persons, 4 scenes |
| Weizmann | 10 classes: walk, run, jump, gallop sideways, bend, one-hand wave, two-hands wave, jump in place, jumping jack, skip | 90 | $180 \times 144$ | frontal/side | Homogeneous outdoor backgrounds, provides irregular versions (with dog, occluded, with bag, *etc.*) |
| Hollywood | 8 classes: answer phone, get out car, handshake, hug person, kiss, sit down, sit up, stand up | 430 | $240 \times 500$ | Various perspectives | Short sequences from 32 movies, cluttered background and occlusions between persons |
| UCF sports | 9 classes: diving, golf swinging, kicking, lifting, horseback riding, running, skating, baseball swinging, walking | 182 | $720 \times 480$ | Various perspectives | Collected from various sports in broadcast television channels, high intra-class similarity between certain classes. |
| UCF YouTube | 11 actions: basketball shooting, cycling, diving, trampoline jumping, golf swinging, horse riding, soccer juggling, volleyball spiking, swinging, walking and tennis swinging. | 200 | $240 \times 320$ | Various perspectives | Still and moving cameras, cluttered background, low resolution, scale, viewpoint and illumination variations. |
| Rushes | Drama productions in the following genres: detective, emergency, police, ancient Greece and historical London. | 109,135 | $352 \times 288$ | Various perspectives | Contains low-quality streams, colour bars, blank screens, complex scene of people, vehicles, buildings, equipments and other objects, recorded both in indoors and outdoors. |
| Flickr | Three entities including 15 objects , 5 locations, and 3 people /characters. | 74,958 | $640 \times 360$ | Various perspectives | Contains a wide range of scenes and viewpoints in complex environments |

Table 2.1: A summary of the datasets used in the experiments throughout this thesis. The comparison involves the number of the actions performed in each dataset, the total number of video clips, frame resolution and finally the characteristics of the frames and actions performed.

$s1$, outdoors with scale variation $s2$, outdoors with different clothes $s3$ and indoors $s4$ as shown in Figure 2.2. It consists of 600 videos sampled at 25 fps (frame per seconds) with the frame size of $160 \times 120$ pixels. The background is homogeneous and static for


Figure 2.2: Examples from the six actions presented in the KTH dataset performed by 25 subjects in four scenarios over homogeneous backgrounds with a static camera. Each column represents one of the six actions: walking, jogging, running, boxing, hand-waving and hand-clapping, while each row represents one of the four different scenarios: outdoors s1, outdoors with scale variation s2, outdoors with different clothes s3 and indoors s4. Figure taken from [Schuldt et al. (2004)].

most cases, containing a single identifiable object. It is considered to be a standard dataset for evaluating action detection and recognition approaches. This dataset was used to evaluate the first stage (Chapter 3) and second stage (Chapter 4).

## Weizmann

The Weizmann dataset was provided by Blank et al. (2005) with ten categories of human actions performed by nine actors: walking, running, jumping, gallop sideways, bending, one-hand-waving, two-hands-waving, jumping in place, jumping jack, and skipping. Each clip lasts about 2 seconds and contains a single person performing one action (sometimes two) resulting in 93 video sequences recorded outdoors and sampled at 25 fps with an image frame size of 180×144. Similar to the KTH dataset, this benchmark is a standard one with a homogeneous and static background. Figure 2.3 displays some frames extracted from this dataset. It was used to evaluate the second stage (Chapter 4).

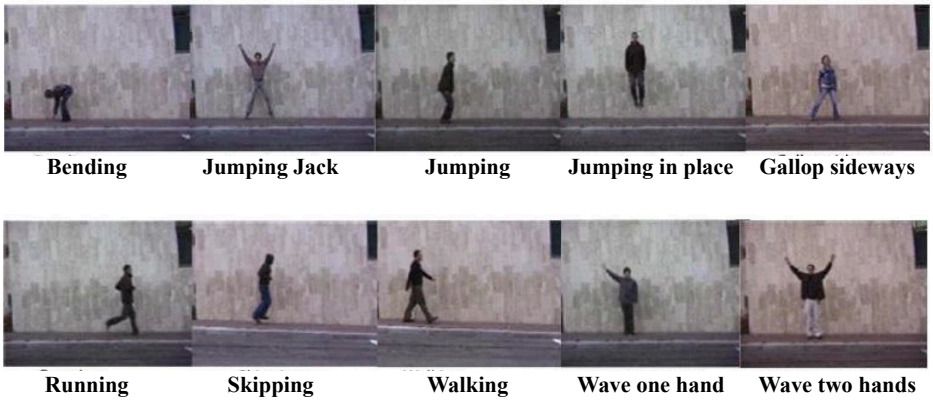



Figure 2.3: Examples of the ten actions in the Weizmann dataset performed by nine subjects over homogeneous outdoor backgrounds. The first row shows five actions: bending, jumping-jack, jumping, jump in place on two legs, gallop sideways, while the second row shows the remaining actions: running, skipping, walking, wave one hand, wave two hands. Figure taken from [Blank et al. (2005)].

## Hollywood Human Action

The Hollywood dataset [Laptev et al. (2008)] contains samples from 32 real-world movies with human actions in eight categorises: answer phone, get out car, handshake, hug person, kiss, sit down, sit up, and stand up. This is a challenging dataset due to its dynamic background, camera motions and variety of interesting actions. Laptev et al. (2008) divided the dataset into a testing set of 211 clips collected from 20 movies and two training sets: '*automatic*' created by script-based action annotation, and '*clean*' labelled manually. They trained the classifier using the *clean* training set containing 219 clips. Figure 2.4 shows some samples from the Hollywood dataset. The dataset, which is widely used to evaluate algorithms developed for the action recognition task and is considered more challenging than the KTH and Weizmann datasets, was used to evaluate the second stage (Chapter 4).

## UCF Sports Action

For more complex and challenging content with different viewpoints and backgrounds, the UCF sports dataset was employed. It was collected by Rodriguez et al. (2008) from sport broadcasting videos and contains ten human actions, but the pole vaulting action is not publicly available. Therefore the approach in this study was applied to the nine available actions consisting of diving, golf swinging, kicking, lifting, horseback riding,


Figure 2.4: Example of the eight actions in the Hollywood human action dataset containing cluttered background and occlusions between subjects. The first row shows four actions: kiss, sit down, sit up, stand up. The second row shows the remaining actions: answer phone, get out car, handshake, hug person. Figure taken from [Laptev et al. (2008)].


Figure 2.5: Some examples of the actions provided by the UCF sports dataset, which contains nine different actions collected from sport videos in a wide range of scenes and viewpoints. Actions in this data set include (first row) diving, golf swinging and kicking; (second row) horseback riding, running, skating; and (last row) lifting, swinging and walking. Figure taken from [(Rodriguez et al., 2008)].

running, skating, swinging and walking. There are approximately 150 videos showing a large intra-class variability (cf. Figure 2.5). The UCF sports dataset contains moving backgrounds and several objects. Therefore it is usually applied to prove the approach's ability to detect, represent and recognise video events. This dataset was used to evaluate the first stage (Chapter 3) and second stage (Chapter 4).

## UCF YouTube Action

For the UCF YouTube Action (UCF11) dataset clips were collected from typical YouTube videos[2] with large variations in terms of object appearance and pose, viewpoint, scale, *etc.*[Liu et al. (2009b)]. Since they were collected from unconstrained videos taken from the web with non-static backgrounds, low-quality camera motions, poor illumination conditions, *etc.*, it is considered more complex and challenging than the four datasets mentioned above. The UCF11 consists of 1,600 videos with 11 action categories: basketball shooting, cycling, diving, trampoline jumping, golf swinging, horse riding, soccer juggling, volleyball spiking, swinging (by children), walking with dogs and tennis swinging. Each category contains 25 scenes with at least four clips for each scene. The video clips in the same category may have similar characteristics such as the same actor, similar background, or similar viewpoint. This dataset was used to evaluate the developed application (Chapter 6). Figure 2.6 shows a samples from the 11 actions presented in this dataset.

## Rushes Video

The TREC video (or TRECVID) is an annual conference series started in 2001. It is sponsored by the National Institute of Standards and Technology (NIST) [Over et al. (2008)]. Every year challenging tasks and a large collection of testing and query data are provided to research groups to encourage content analysis and information retrieval. TRECVID 2011 was the last version with testing and development data produced from the rushes collection. Rushes are unstructured video sequences with heterogeneous contents. They occasionally contain frames such as low-quality audio/video streams, colour bars, and blank screens that will not be used for the final production. The unedited footage contains a complex scene of people, vehicles, buildings, equipment and other objects, recorded both indoors and outdoors. Repetitive content is often scattered over the entire sequence due to multiple retakes of the same

[2]http://www.youtube.com


Figure 2.6: Examples of actions provided by the UCF YouTube Action dataset, which contains 11 different actions collected from the YouTube website with various variation. Column 1: samples from basketball shooting, cycling, diving, trampoline jumping, golf swinging and horse riding. Column 2: examples from soccer juggling, volleyball spiking, swinging by children, walking with dogs and tennis swinging. Figure taken from [(Liu et al., 2009b)].

scenes. The same objects (e.g. a person) appear in the same or different scenes and settings (e.g. clothing, location, dialogue, camera angle). The content management task for rushes is highly challenging. The usability of each retake is a concern during production, resulting in numerous retakes with repeated contents. It was observed that the length of raw footage was sometimes as much as 40 times longer than its final production [Over et al. (2008)]. Retakes are not copies. They are made because of, *e.g.* , incorrect performance by actors, some missing objects in a scene, or low-quality sound. A successful take may also be repeated for backup purposes.

The contents commonly seen in rushes are:

1. Usable scenes: successful takes that can potentially be used in final production;

2. Camera adjustment: not for a final production;

3. Clapper-board screen: a clapper-board records the scene and the number of the retake that follows;

4. Irrelevant subsequence: *e.g.* colour bar and mono colour screens (perhaps due to faulty recordings).

Figure 2.7 presents some frames from the BBC rushes video dataset. This thesis used part of the 2008 collection, that contains 200 hours, to evaluate the third stage (Chapter 5) and the developed application (Chapter 6). The task was the identification of repeated contents as part of the video similarity task regardless of what these contents are. The problem of indexing or searching any types of repetitions is not addressed. Further, scene segmentation was not operated.

## Flickr dataset

The instance search (or INS) task was one of the TRECVID tasks that involved searching for a recognizable entity – a person, object or place – in a video dataset, given a still image as a query [Over et al. (2008)]. For each entity a number of topics are defined, for example "Barack Obama" as person, "white house" as place and "red car" as object. The task aims to locate for each query image up to a specific number of clips that most likely contain a recognisable instance of the topic. In 2012 data was derived from the Flickr website [3]. A set of 91 video clips were decomposed into short clips of roughly equal length. There were more than 70,000 test video clips and 21 recurring queries of still images. Some transformations were also applied randomly on the test clips.

Inspired by this task, a task is formulated in Chapter 6 to deal with video INS based on a video stream as a query. For the three entities (place, person and object) provided by NIST, three specific topics for location were chosen ('*Eiffel Tower*', '*White House*' and '*Stonehenge*'), three specific topics for persons ('*singer*', '*broadcaster*' and '*actor*') and three specific topics for objects ('*bridge*', '*car*' and '*London tube*'). From the whole collection, we selected six video clips for each of nine classes, of which two were used as a query and the rest were kept for evaluation. Twelve additional video clips were picked up randomly for each entity (*i.e.* a total of 36 clips), making 18 short videos in the query set and 72 short videos in the test dataset. Sample screen shots from each entity with their topics are shown in Figure 2.8.

[3] https://www.flickr.com

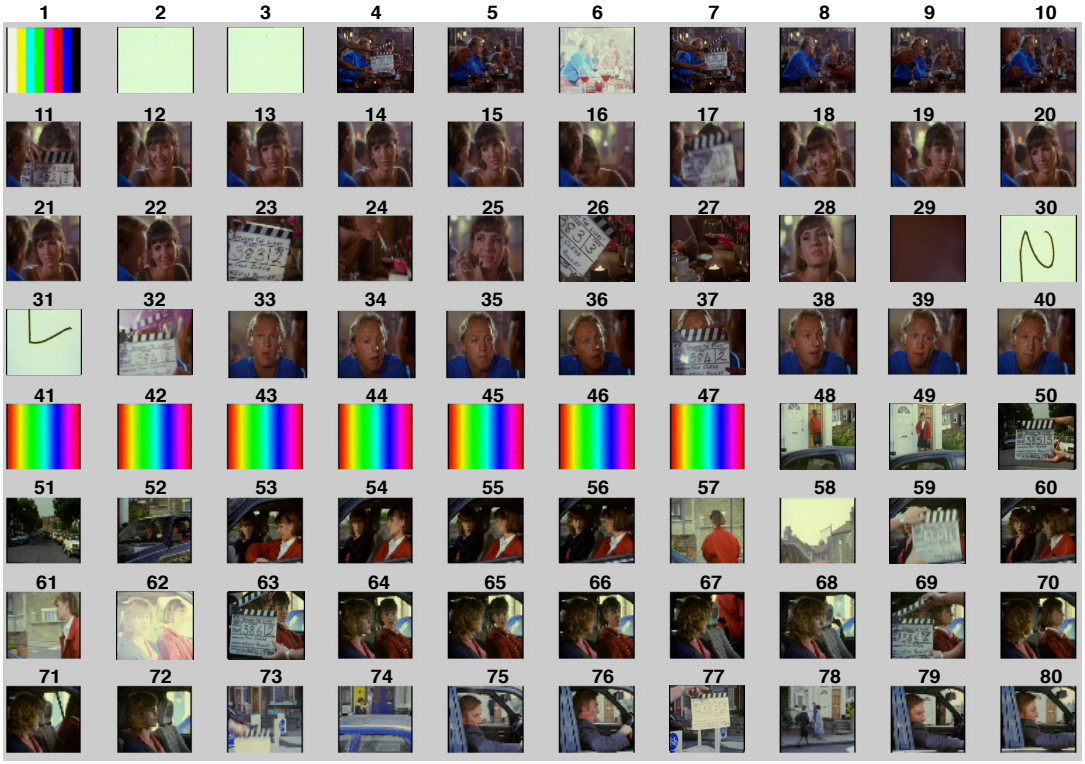


Figure 2.7: A 14-minute sample video identified as "MS212890" from the BBC rushes video collection, consisting of three actors performing in indoor and outdoor scenes. For illustration purposes, video frames were sampled at 0.1 fps (one frame in every 10 seconds) ordered from left to right, then from top to bottom. The sequence contained a number of different types: blank screen (e.g. frames 2–3, 29, 30–31), colour bars (frames 1, 41–47), low-quality video frames (e.g. frames 6, 62), clapper-board (e.g. frames 11, 23), and finally usable scenes (e.g. frames 12–15, 51–57). There were nine retakes (frames 4–6, 7–10, 11–16, 17–22, 23–25, 26–28, 32–36, 37–40) for the indoor scene and six retakes (frames 50–58, 59–62, 63–68, 69–72, 73–76, 77–80) for the outdoor scene. Different camera angles resulted in a different set of actors focused for the same scene (frames 12–16 vs 33–36). The final sequence would be created by a video editor.

Figure 2.8: Screen shots from the Flickr clips. Columns 1–3 represent the entity place with three topics: '*Eiffel Tower*', '*White House*' and '*Stonehenge*'. Columns 4–6 show the entity person with three topics: '*singer*', '*broadcaster*' and '*actor*'. Columns 7–9 are from the entity object with three topics: '*bridge*', '*car*' and '*London tube*'.

## 2.5 Applications

Providing a tool for video sequences alignment can be a basic step in several tasks such as indexing, parsing, action recognition, searching and retrieval. Successful application in the development stage to a complex data sequence such as the rushes video gives confidence that the presented approaches can be practical for various real-life video applications.

Since the main focus of this thesis is the video sequences alignment in video similarity domain, three applications were chosen to validate the performance of the framework. Starting from the standard application in the video similarity domain, the first one is a video clip retrieval task. It can be considered as the main application to evaluate the video similarity technical and can be extended to different applications such as retrieving similar actions for the human action recognition task. The many-to-many graph-matching method from the image-processing domain was extended in this application to measure the similarity between two spatio-temporal graphs in the lower-dimensional space. One step foreword, the second application is to rank the retrieved results from the retrieval application. This step makes the generated results more valuable for various applications such as video search engine. For that a synchronisation map is calculated between the trajectories of two video clips to discover the underlying structure in the manifold. The third application is the more challenging and advance one in video similarity domain. It is been chosen to measure the framework performance in defining more accurate results which can be used to develop a

video search engine using a video query. INS based on video queries that aim to locate clips with a specific entity in a collection of test video clips. For that, each manifold is represented by a set of subspaces; the distances between a pair of subspaces (each from one of the manifolds) is then found.

### 2.5.1 Video Clip Retrieval

**The task.** This application evaluates the proposed framework in the video clip retrieval task. Using the UCF YouTube action dataset, a smaller collection was created with 550 video clips. From the 550 videos, five query clips were randomly picked for each one of the 11 actions, making 55 videos in the query set. Given that a query clip contains one of the 11 actions, the task is to define all the related clips with the same action category from the whole collection of 550 videos.

**The application.** Instead of one-to-one matching between two sets of features or each pair of frames, many-to-many matching methodology between two graph-based representations was adopted. As shown in Figure 2.9, the similarity problem between two clips is formulated as graph matching in two stages. In the first stage, a bipartite graph with spatio-temporal neighbourhood is constructed to explore the relationship between the data points and estimate the relevance between a pair of clips. In the second stage the problem of matching between a pair of video clips is converted to compute the minimum cost of transportation within the spatio-temporal graph. This application demonstrated the retrieval capability of the proposed framework compared to the existing methods.

### 2.5.2 Video Clip Ranking

**The task.** One of the interesting topics in computer vision applications is the detection of nearly-repetitive contents within video sequences. Given a video stream with repetitive contents, the problem to be addressed is how to define the relation between nearly-repetitive sequences in a low-dimensional space. A collection of video sequences were selected from the NIST TRECVID 2008 BBC rushes video collection [Over et al. (2008)]. Each video contains a set of scenes located by a line of actions provided by NIST and was used as a unit for evaluation. These description were used to define the position of the scenes and the retakes, which were used as the ground truth. The purpose of the experiment was to retrieve multiple similar retakes of the same scene in a ranked list. More details are presented in Chapter 6.

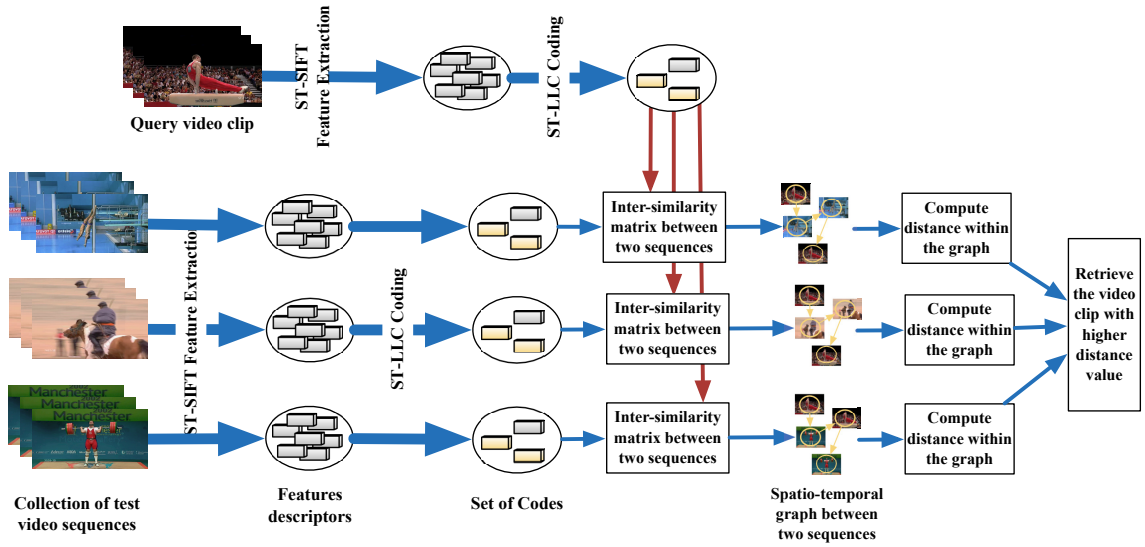



Figure 2.9: Overview of the video clip retrieval application developed to evaluate the video sequences alignment framework. It measures the similarity between two clips by matching their spatio-temporal neighbourhood graph, which is constructed as part of their representation. A graph distance measurement is applied to compute the similarity and retrieve the video with higher value as the most similar.

**The application.** This application solves the ranking problem by embedding the multiple video retakes of the same scene to a lower-dimensional space that preserves the intrinsic geometry of the data. Given a query clip, the framework explores the relationship between the data points and estimates the relevance between the query and remaining retakes in the database. The similarities observed in frame sequences are captured by defining two types of correlation graphs: an intra-correlation graph within a single sequence and an inter-correlation graph between two repetitive sequences. A synchronisation map between two clips is then calculated using the manifold embedding. Figure 2.10 briefly describes the concept of this application. The application shows that the proposed framework attains a significant improvement on embedding repetitive sequences over the conventional methods.

### 2.5.3 Instance Search Based on Video Queries

**The task.** This application was inspired by the TREC Video INS task to address the problem of matching video clips, each of which contains an instance belonging to the same entity but undergoing a transformation such as variations in viewpoint and changes in lighting conditions. The original INS task is to search for a recognizable

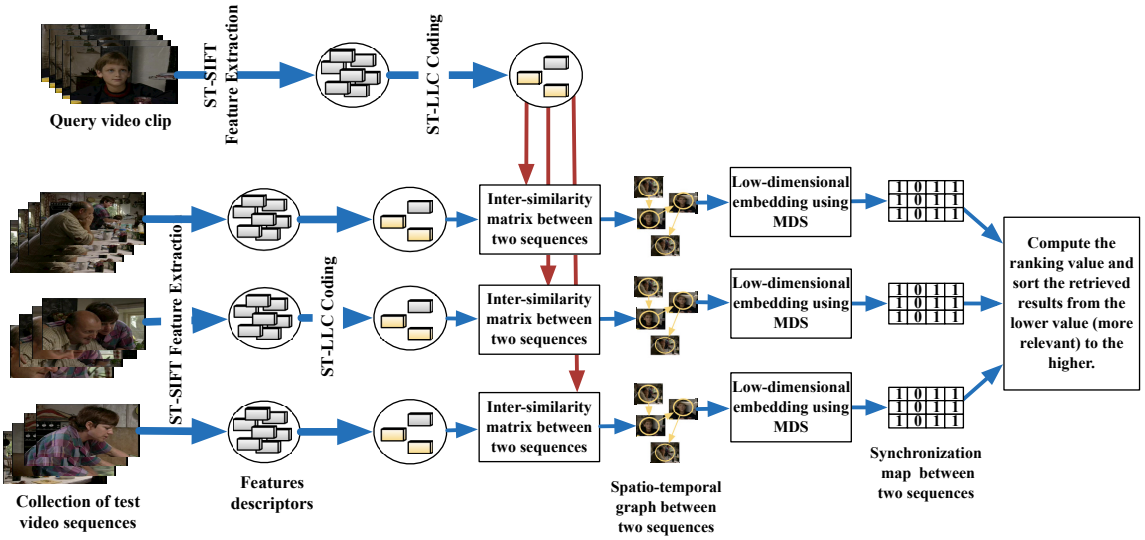



Figure 2.10: An overview of the video clip ranking application developed to evaluate the video sequences alignment framework. The video representations for both query and testing data are defined as trajectories in the lower-dimensional space. A synchronisation map between two clips is then calculated and the ranking values are computed. A list of retrieved videos is returned sorted based on the ranking value.

person, object or place in a video dataset, given a still image as a query. The task was adapted to deal with video INS based on a video stream as a query. From the 2012 TREC Video Flickr data collection, a small dataset was defined and nine query clips were chosen, containing one of three entities – place, person or object – and each with different 'topics'. Given a collection of test video clips and a collection of query video clips containing a person, an object, or a place, the proposed method aims to locate for each query up to the specific number of clips that most likely contain a recognisable instance of the entity (referred to as an topic). More details are presented in Chapter 6.

**The application.** The problem was formulated as manifold matching, *i.e.* measuring the similarity between multiple manifolds, each of which represents a video clip. The video contents were analysed by characterising the spatio-temporal information embedded in a frame sequence. The spatio-temporal neighbourhood graph is firstly applied as video representation. Linear models are then extracted using a clustering method. Each video clip is then modelled as a manifold with a set of subspaces, and the task becomes finding the distances between a pair of subspaces, each subspace from one of the manifolds. This application shows an improvement in the search and retrieval performance over conventional approaches, implying that the proposed

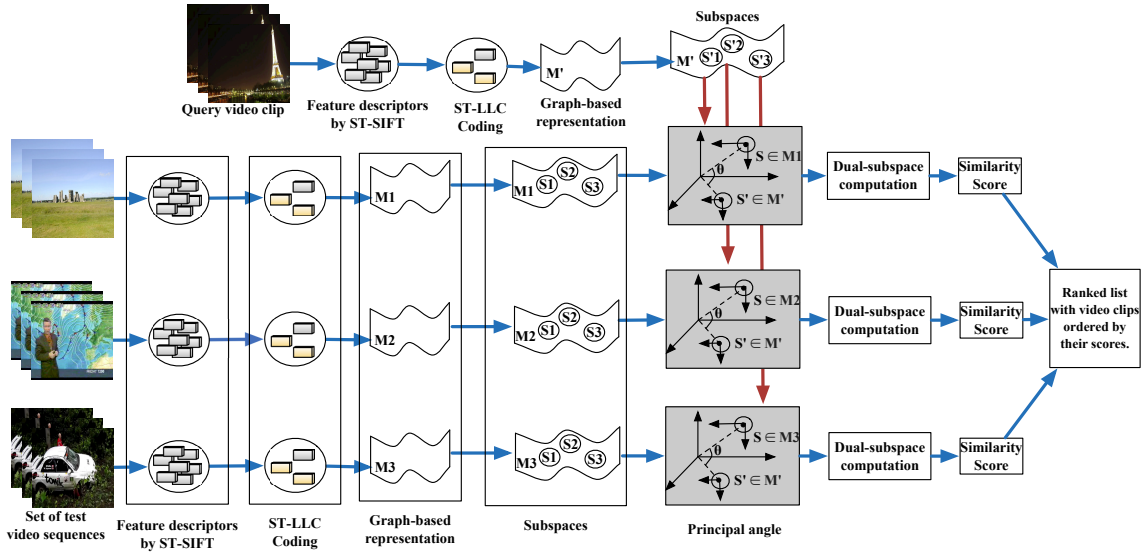



Figure 2.11: The steps involved in the instance search application developed to evaluate the video sequences alignment framework. The problem is solved by matching a pair of manifolds, each represented by a set of subspaces. The video representation is firstly defined in the lower-dimensional space. Then linear models are extracted from each manifold and grouped into subspaces. The similarity is then measured between a pair of subspaces, each from subspace from one of the manifolds. A ranked list is created with video clips ordered by their similarity.

framework can be practical in more complex applications. Figure 2.11 illustrates the concept of this application.

## 2.6 Summary

This chapter has presented a general overview of the developed video sequences alignment framework that utilises all the proposed algorithms, as well as the dataset behind this framework and other experiments defined throughout the thesis. A more functional description of stages is provided in the following chapters: Chapter 3 covers the first stage related to feature extraction method, Chapter 4 presents the coding technique as the second stage, while Chapter 5 is related to the manifold embedding stage. Three applications, related to the video similarity tasks used to evaluate the framework, were also introduced in this chapter and more details of the evaluation results form the content of Chapter 6.

# Chapter 3

# Stage 1: Space-time Video Representation

Chapter 2 presented an overview of the video sequences alignment framework with three stages containing different types of features. This chapter presents the first stage which contains ST-SIFT as a space-time extension of SIFT detector originally applied to the 2D volumetric images. Most previous extensions dealt with 3D spacial information using a combination of a 2D detector and a 3D descriptor for applications such as medical image analysis. In this work, a spatio-temporal detector was defined based on the 3D-DoG pyramid to detect the local extrema. Interest points were extracted not only from the spatial plane ($xy$) but also from the planes along the time axis ($xt$ and $yt$). The space-time extension was evaluated using the human action classification task. Experiments with the KTH and the UCF sports datasets show that the approach is able to produce results comparable to the state-of-the-art techniques.

The chapter is structured as follows. Section 3.1 introduces the video representation and the action recognition problem as the task used to evaluate this stage, along with the motivations of this extension and its contributions. Section 3.2 reviews previous work related to the spatio-temporal feature extraction task and to the previous extensions of SIFT, and defines the spatial SIFT. The implementation of the proposed ST-SIFT detector is presented in Section 3.3. A summary of the results obtained from the evaluation experiment is given in Section 3.4. Section 3.5 provides a concluding discussion.

## 3.1 Introduction

Representing and interpreting video content has received great attention in computer vision applications. Video volume has a highly important information about changes in the environment and is crucial for various video-processing tasks such as surveillance and video indexing [Laptev (2005)]. Video structure consists of a set of images that are not restricted to certain velocity or appearance over the time domain. On the other hand, video events have strong variations in both spatial and temporal dimensions. Local spatio-temporal features have been extensively investigated as sparse and compact representations of video content and have proved their ability in different computer vision applications. These features are defined from the key points that contain a sudden change over both space and time domains. An effective interest points detector is important for the delivery of a compact video representation as a core component for many applications such as human action recognition.

Detecting humans and their motions in a video stream is a core task for most of the feature extraction techniques, and it is considered here to define the performance of the development work as well as to locate the rough position of the proposed method among the state-of-the-art techniques. The task aims to define the actions and goals of one or more objects from a set of observations on the object's activities and the environmental changes [Doctor et al. (2011)]. The main elements in this process are: the actions (which are performed by an object and cause changes to the environment) and goals (which are the sets of changes to the status of the environment that the object aims to achieve). The start of this research field in the 1980s captured the attention of many computer science communities because of its vital role in supporting many different applications and its connection to various fields of study such as medicine, human-computer interaction (or HCI), and sociology.

Various interest point based approaches have been introduced to represent video events [Dollar et al. (2005); Niebles et al. (2008); Schuldt et al. (2004)]. These methods consist of two steps: a feature detection phase that searches for the interest points through the frames, followed by a feature description phase that describes the area around the detected points and then defines a model based on independent features such as BoW [Sivic et al. (2005)] or based on structural information such as sparse coding (SC) [Olshausen and Fieldt (1997)]. Most previous works, especially on the action recognition task, used the BoW model as a feature representation, and proved robust to location changes and to noises. However, they depended mainly on the descrip-

tor phase to produce discriminative representation of a video, discarding information relevant to the distribution of interest points in the spatio-temporal domain. As a consequence, the features produced often lacked temporal information for describing smooth motions. Furthermore, they were not able to address the scale and location invariance in the temporal domain.

### 3.1.1 Motivations

To represent video events, high-level semantic features such as texture, colour or shape have been obtained either by textual annotation or by complex procedures based on visual content [Long et al. (2003)]. Measuring the similarity between these types of features is expected to be done in a way that human beings would perceive or recognise. Even for exactly the same inputs, different users would have different interpretations of the similarity. Therefore, low-level features were mostly used as video content representation. Multiple video sequence feature detectors extended established 2D interest point detectors to the spatio-temporal domain. These extensions took the 3D nature of video data into account and extracted features from space and time domains. Some of them extracted ST features from small video patches (named cuboids) [Dollar et al. (2005); Laptev et al. (2008)]. However, cuboid sizes first need to be defined, a tough task, and they usually give a low number of features which is insufficient for most part-based classifiers. Other works tracked the interest points along the video sequence and then determined the motion information from the points' trajectories [Jain et al. (2013); Wang et al. (2013)]. These approaches do not scale well with a large number of local regions and become computationally expensive for large-scale video datasets.

In the literature, scale invariant feature detectors, such as the Harris-Laplace detector [Mikolajczyk and Schmid (2004)] and SIFT [Lowe (2004)], have played vital roles in vision applications. The key idea is that at each interest point, one or more scales are selected based on a specific scale-invariant function such as the Laplacian of Gaussians. The hypothesis here is that local extrema of this function occur at the same scales for the same interest point in different images. This allows the extracted features to be matched between images with different scales [Morel and G.Yu (2011)]. A typical image, generally, often has relatively few pixels selected from reliable scales. Therefore, finding correspondences between scale invariant features has so far been applied mainly to a few pixels in each image. Since its original development by Lowe (2004), SIFT has been successful employed in various image-processing applications for locally

detecting and describing interest points. It has proved its efficiency with tasks in a 2D space such as image similarity and classification. Its extension to higher-dimension spaces has been explored in order to represent more complex data and overcome the matching problems. However, most of these works extend the algorithm spatially to represent the 3D data, for example in medical images.

This chapter describes a way to alleviate the shortcomings of the existing ST approaches: sensitivity to scale, location and rotation changes; the low number of features to represent the video; the need for the pre-processing steps applied before feature extraction. The aim is to get the benefit of SIFT and extend its efficiency in video-processing applications. We propose the ST-SIFT approach to detect and represent interest points in video streams. These points capture both appearance and motion from a region of interest at multiple scales, which makes them invariant to different changes. This work tackles two important issues that have not been addressed in the previous extensions made for SIFT. One is the transformation of video signals into 3D spatio-temporal pyramids that deal with both the 2D space and additional time domain simultaneously. The other is the extraction of interest points both from the traditional spatial plane and from the temporal plane. These lead to detection of invariant local regions to scale and location not only in the spatial domain but also in the temporal domain.

### 3.1.2 ST-SIFT: Overview

This chapter presents a SIFT extension for detecting interest points that can have significant local variations in both the spatial and temporal domains. ST-SIFT was inspired from two previous studies: the first one by Dorr et al. (2010) in which spatio-temporal Laplacian pyramid was constructed as a multi-resolution representation for video streams. This pyramid was applied to help visualise dynamic gaze density maps. Using this as a starting point, this study transforms the 3D (2D space and time) video volume into spatio-temporal Gaussian and DoG pyramids with multiple scales. The second one, presented by Lopes et al. (2009) for human action recognition application, collected 2D SIFT and 2D SURF (speeded-up robust features) interest points on the $xy$ plane along the spatial domain and on the $xt$ and $yt$ planes along the spatio-temporal domain. Inspired by that, this study segments the constructed pyramid into spacial and spatio-temporal planes and then detects the common features between them. To describe the region around the detected points the 3D SIFT descriptor developed by Scovanner et al. (2007) was used. This calculates the spatio-temporal

gradient for each pixel in the given patch. The approach leads to local regions that are invariant to scale, location and orientation in both the spatial and the temporal domains. The human action classification task was chosen for evaluation, because it is the task which contains reported performances from most of the new developed methods, and therefore data for comparison purposes.

### 3.1.3 ST-SIFT: Contributions

The main contributions of the proposed work fall into the following aspects.

- Construction of multi-resolution space-time Gaussians and DoG pyramids, where each level contains a 3D smoothed and subsampled version of the previous level.
- Provision of an interest point detection schema from three different planes along the spatial and the temporal axes.
- Formulation of the space-time detector that is scale and location invariant.
- Application of the developed ST-SIFT on a human action classification task, with comparison to state-of-the-art approaches.

## 3.2 Related Work

As a part of the scope of this thesis is feature representation in video sequences, existing methods based on the type of features used to model the video events are also reviewed. Existing methods of video representation can be categorised into two classes: local and global representations.

Global representations, or holistic methods, deal with video sequences as a whole signal rather than as patches or regions [Poppe (2010)]. These kinds of representations are obtained in a top-down fashion, where the object is firstly detected using pre-precessing steps and then the features are directly encoded as a whole. They have been successfully employed for different applications, because of their ability to encode more visual information by retaining the spatial and temporal structures of the video. However, they are sensitive to noise, occlusions and background variations. Also they depend on preliminary steps, such as background subtraction, segmentation and tracking, which are computationally expensive and time-consuming in some complex scenarios.

Local representations are used to encode video events as a set of local spatio-temporal features or descriptors [Zhen (2013)]. These features are based on extracted spatio-temporal interest points (STIPs) from video sequences. The process of local representations follows a bottom-up fashion: spatio-temporal interest points are first extracted, local patches around these points are then calculated, and finally the final representation is defined by combing the patches. In contrast to global representations, the local detectors avoid pre-processing steps, and they are less sensitive to noise and partial occlusion. However, they are by definition local and their success depends on extracting a sufficient number of interest points.

In the following sections, we focus on the local representation since it is the core of this chapter. Section 3.2.1 discusses the recent literature (but does not pretend to give complete coverage to all works). Section 3.2.2 introduces the original algorithm of the 2D SIFT which is a spatial local representation extended in this work to the spatio-temporal domain, and follows with a review of extensions accomplished previously.

### 3.2.1 Local Space–time Video Features

Local video representations define an observation as a collection of local detectors and descriptors [Poppe (2009)]. Accurate feature localisation and background subtraction are not required for these types of representations and they are invariant to changes in viewpoint, person appearance and occlusions. The process in these representations involves detection and description. Feature detectors maximise specific saliency functions to detect spatio-temporal features with locations and scales invariant in the video, and we discuss these in Section 3.2.1.1. Feature descriptors measure spatial or spatio-temporal image gradients and optical flow to describe the shape and motion in the surrounding regions of the detected features, and we discuss these in Section 3.2.1.2.

#### 3.2.1.1 Detectors

Spatio-temporal feature detector has been heavily employed with much success in content-based video analysis. To extract the spatio-temporal features, several approaches have been proposed to extend the 2D features to the temporal dimension. Laptev and Lindeberg (2003) were the first to propose such a spatio-temporal extension, based on the Harris–Laplace detector proposed by Mikolajczyk and Schmid (2004). They sparsely detected interest points using a time-consuming iterative

procedure applied for each feature candidate separately. Therefore, by detecting a small number of interest points to keep the computation time under control, the local neighbourhood has significant variations in both spatial and temporal domains. Oikonomopoulos et al. (2005) presented a spatio-temporal extension of the salient region detector defined by Kadir and Brady (2001). The entropy within each cuboid was calculated, then the centres of those with local maximum energy were selected as salient points. The entropy values of each salient point were then maximised to define the scale. Their features are scale-invariant but sparse similar to the original spatial detector. Schuldt et al. (2004) presented a video representation using 3D space-time interest points [Laptev and Lindeberg (2003)] detected from each video. The detected points are clustered to form a dictionary of video words. Each action sequence is then represented by a BoW. A classifier is then learnt for each action using the support vector machine (SVM) classification schema. Campos et al. (2011) used spatio-temporal shapes (STS) for action recognition, where the action is analysed as a single 3D shape in the spatio-temporal cube.

Willems et al. (2008) extended the 2D Hessian detector and SURF descriptor to the spatio-temporal domain. Saliency were identified as the determinant of a 3D Hessian matrix, which is calculated using integral videos. Gilbert et al. (2009) proposed a fast multi-action recognition algorithm using a data mining techniques to find the frequently occurring patterns on dense 2D Harris corners. Noguchi and Yanai (2012) extended the 2D SURF features [Bay et al. (2006)] to the spatio-temporal domain and combined this with Lucas-Kanade optical flow [Lucas and Kanade (1981)] as a feature detector. Their features were extracted from the moving SURF keypoints, which failed to deal with some videos containing camera motion or holistic decision of motion thresholds in the selection of interest points. Nga and Yanai (2013) solved this using the Delaunay triangulation to model the relationships between interest points along the video sequence. To capture the geometrical distribution of interest points, Yuan et al. (2013) firstly extracted the 3D Harris interest points and followed by applying the 3D R transform defined as an extended 3D discrete Radon transform, which is invariant to geometrical transformation and robust to noise. To reduce the dimensionality of the 3D feature matrix, they applied the PCA, obtaining the R features. As an encoding phase, they used the BoW representation. Then, a fusion method was used to fuse these two features.

A common drawback of these approaches is the small number of stable interest points available, which make them unable to cope with changes in space or time.

This problem was addressed by Dollar et al. (2005). They presented an approach for behavior recognition motivated by the ideas of Laptev and Lindeberg (2003) and Schuldt et al. (2004). The local maxima were extracted from space and time domains based on the responses of the 2D Gaussian filter convolved with a quadrature pair of 1D Gabor filters. However, the generated features are not scale invariant, sensitive to illumination change and the size of the cuboids needs to be determined by the user. In a similar approach, Rapantzikos et al. (2007) use the responses obtained by applying a discrete wavelet transform (DWT) in each of the three directions ($x$,$y$,$t$) of a video volume. Different combinations can be obtained by using either a low-pass or a high-pass filter for each dimension. Each of the generated sub-bands corresponds to slow or fast characteristic movement in one of three dimensions. The responses in these sub-bands are then used to extract salient points in space and time. Oshin et al. (2008) extended the semi-Naive Bayesian classifier called Ferns, proposed in Ozuysal et al. (2007), and used it to approximate the behavior of interest point detectors. They learn ferns on the regions around spatio-temporal interest points extracted using a cuboid detector approach. Later, Le et al. (2011) introduced independent subspace analysis (ISA) as an unsupervised deep learning algorithm. It learns spatio-temporal interest points extracted from unlabelled videos. They presented high performances using different datasets; however, a very large number of parameters need to be adjusted and the algorithm also needs a large number of training samples, which restricts its applicability. Weinland et al. (2010) defined a global classification decision using 3D-HOG (histograms of oriented gradients) descriptors followed by a hierarchical classifiers. A classifier was then learn from training examples with various views to handle viewpoint changes. More recently, Lu and Aggarwal [Xia and Aggarwal (2013)] combined the 1D Gabor and the 2D Gaussian filters to extract spatio-temporal features followed by a cuboid similarity feature to describe region around these features.

Instead of extracting the features over the entire video sequence, Wong and Cipolla (2007) detected subspaces of correlated movement which corresponded to large movements by the object and were used to detect a sparse set of interest points. Their work required some pre-processing to convert the input video into samples containing one action each. Similarly, Bregonzio et al. (2009) calculated frame differences to define the regions with focus of attention. Then 2D Gabor filtering was applied with different orientations to detect salient points within these regions. Ning et al. (2007) defined histogram representation by firstly convolving the video signal with a bank of 3D Gabor filters, then pooling a limited number of the responses through a MAX-like

operation. The histograms were then generated by quantisation of the orientations in nine directions, which is the final patch-based feature. Wu et al. (2011) captured the motion using Langrangian particle trajectories, which are dense trajectories defined by numerical integration of optical flow over time. These trajectories are then decomposed into camera-induced and object-induced components, which help to handle camera motion.

Silhouettes have also been used to classify actions, based on the assumption that that human activities can be represented as a continuous progression of the body posture. Sun et al. (2011), for instance, represented each frame in the video with a self-similarity matrix (SSM) that define a feature vector. A combination of all SSMs is then created and decomposed into its rank-1 approximation, resulting a set of compact vectors with high ability to classify different actions.

#### 3.2.1.2 Descriptors

Local descriptors are used to summarise the regions around interest points in the image or video representation. These regions are mostly invariant to background clutter, appearance and occlusions, and are possibly invariant to rotation and scale. The size of the regions is defined based on the scale of the interest point.

Schuldt et al. (2004) computed spatio-temporal jets to capture the information of the region around the feature. These jets [Poston and Stewart (1978)] are the normalized derivatives of the features with respect to the space and time. Niebles et al. (2008) learnt codewords as an intermediary representation for each action, and then the distribution of these codewords was used to model the class label. The brightness gradients were computed for each word, smoothed, concatenated to form a vector and projected to the lower-dimensional space using PCA. Dollar et al. (2005) combined different local space-time descriptors based on brightness, gradient, and optical flow information. Different descriptor variants were investigated, including normalised pixel values, brightness gradient and windowed optical flow. Moreover, the PCA was applied to reduce the descriptor dimensionality. Overall, this descriptor, which is based on concatenated gradient information, is shown to give a good performance. Yeffet and Wolf (2009) efficiently classify actions using local trinary patterns (LTP), inspired by the local binary pattern (LBP) [Ojala et al. (2002)], combined with a linear SVM classifier.

Region of interest points have also been described by local grid-based descriptors, which summarise the local characteristics within grid cells and ignore the slight varia-

tions in the space and time domains. For instance, Laptev et al. (2008) combined both appearance and motion information in their descriptor using HOGs and histograms of optical flow (HOFs) in a late fusion approach. Klaser et al. (2008) presented a 3D extension for the HOG descriptor, referred to as the histograms of spatio-temporal gradient orientations (3D-HOG). They extended the integral images concept to 3D, which involves a dense sampling of the feature cuboid over multiple scales and orientations in both the space and time domains. Campos et al. (2011) also represented the actions as space-time volumes, followed by feature vectors construction based on the 3D-HOG.

Several approaches combine interest point detectors and descriptors in a feed-forward framework. Jhuang et al. (2007) developed a hierarchical framework with multiple stages, where Gabor filters are applied at the lower level followed by a local max operation. The responses are then used in a higher level with a global max operation. A final matching stage with prototypes defines the final representation. After that, the framework was extended by Schindler and van Gool (2008) to combine both shape and flow responses. Escobar et al. (2009) applied a linear spatial-temporal filtering as a first stage followed by local and global non-linearities known as normalisation. These responses are then used to train a model and build motion maps that define the activities performed in the video.

Dealing with local descriptors in similarity and classification tasks is not straightforward because of their different numbers and their high dimensionality. Thus a codebook is defined as a solution by clustering the region of interest and selecting the cluster centres as representative codewords. A local descriptor is described as a codeword and a video sequence is then represented as a BoW or a histogram of codeword frequencies. For example, Schuldt et al. (2004) represented the video sequence with a collection of visual words of normalised derivatives in space and time. The final representation is the histogram of number of occurrences of specific visual words in the given video. Niebles et al. (2008) modelled each action in the video sequence with a BoW in an unsupervised fashion. The brightness gradients are calculated for each visual word followed by image gradient, which are concatenated and projected onto a lower dimension using PCA as the final representation. Agusti et al. (2014) extended the BoW by to the t-BoW by counting the number of words co-occurring at specific time separations using the frame number as unit. Once the histograms of words computed, a probability is defined and the concatenation of these probabilities form the final representation. Liu and Shah (2008) combined spatio-temporal features

with spin images described by spatio-temporal volume. A matrix of the features occurring and the action videos is then constructed, decomposed into eigenvectors and finally mapped to a lower-dimensional space. Xia and Aggarwal (2013) computed 3D cuboid around interest points which was divided into voxel contains number of pixels, a histogram of the pixels is then computed and normalized to be used later for the similarity.

Beyond the BoW model, Yao et al. (2010) developed a hough transform voting framework. A randomised tree model was utilized to learn the extracted 3D local video patches and then the corresponding votes were defined in the 4D Hough-transformed space. The 3D feature patches were mapped into a 4D votes in the Hough space and the actions were classified using a multi-class classifier based on the leaves of the trees.

### 3.2.2 Scale-Invariant Feature Transform (or SIFT)

Before considering the proposed extension to the 2D SIFT [Lowe (2004)], this section introduces some preliminary work: the original algorithm of the 2D SIFT detector (Section 3.2.2.1); the 2D SIFT descriptor (Section 3.2.2.2); and extensions accomplished previously on the detector and descriptor sides (Section 3.2.2.3).

#### 3.2.2.1 2D SIFT Detector

The 2D SIFT detector maps the spacial content of an image to a coordinate of scale, location and orientation invariant features [Lowe (2004)]. The major steps to detect and describe interest points in the 2D SIFT are illustrated in Figure 3.1.

The first step is to construct the scale-space representation for the image. This is achieved using a scale-space kernel function such as the Gaussian, which is a continuous function to capture stable features in different scales. The Gaussian function on a point $(x, y)$ at scale $\sigma$ can be defined as

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \tag{3.1}$$

The scale-space function $L(x, y, \sigma)$ for the input image $I(x, y, \sigma)$ can be defined as:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y, \sigma) \tag{3.2}$$

where $*$ is the convolution operation (cf. Figure 3.2).

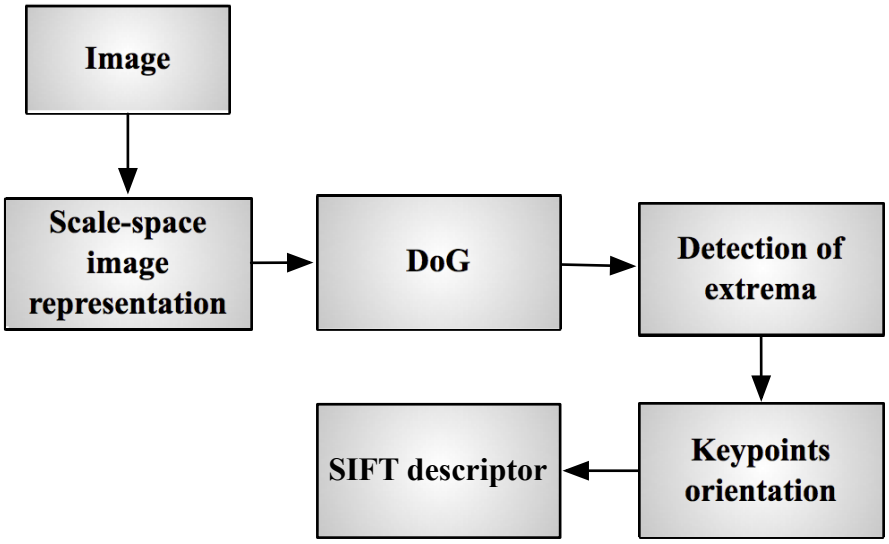



Figure 3.1: The main steps in the 2D SIFT approach.

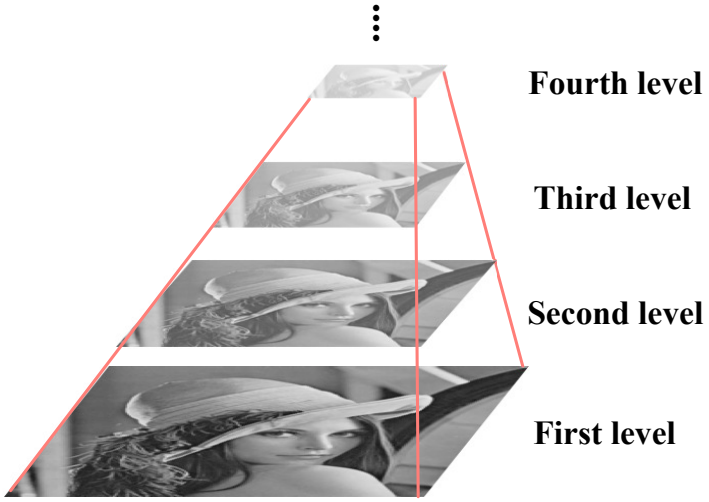



Figure 3.2: Example of a multi-scale representation for "Lena" image constructed by convolving the input image with a Gaussian filter. At each level of the scale pyramid the image is sub-sampled and smoothed by a Gaussian filter.

The second step is to compute the DoG pyramid $D(x, y, \sigma)$, which indicates stable locations in the scale-space, derived at each octave or level by convolution of the input

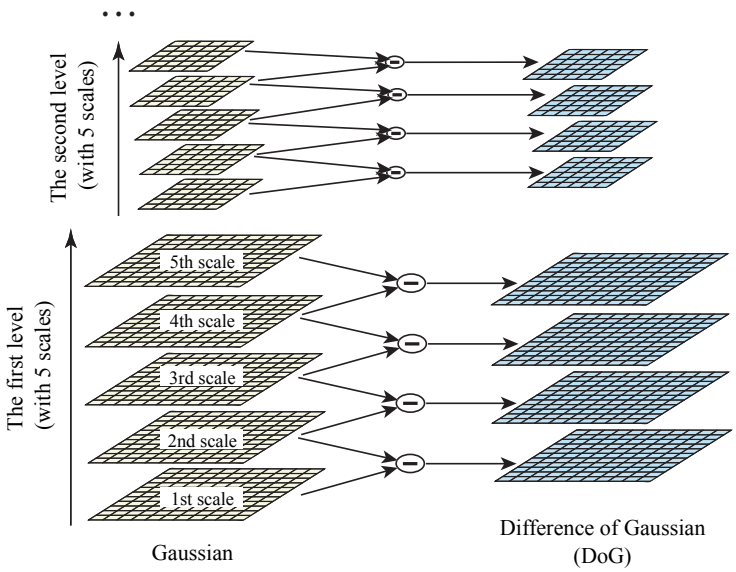



Figure 3.3: The process for constructing the DoG pyramid. For each octave of scale space, the initial image is convolved with Gaussian to define the scale-space image (shown on the left). Then the neighbour Gaussian images are subtracted to generate the DoG (images on the right). Moving from the first octave to the second, the Gaussian image is down-sampled by 2, and the steps are repeated. Figure taken from Lowe (2004).

image with the Gaussian function as following:

$$D(x, y, \sigma) = (G(x, y, K\sigma) - G(x, y, \sigma)) * I(x, y, \sigma) \tag{3.3}$$

$$= L(x, y, K\sigma) - L(x, y, \sigma) \tag{3.4}$$

where the DoG function is the difference between two adjacent images separated by a constant scale factor $K$ (cf. Figure 3.3). To start the following octave, the Gaussian image is sub-sampled by 2, and the steps are repeated.

Then the maxima and the minima of $D(x, y, \sigma)$ give scale-invariant points in the scale-space. These are defined by comparing each point with eight neighbours in the current scale and nine neighbours in the scale above and below, as shown in Figure 3.4.

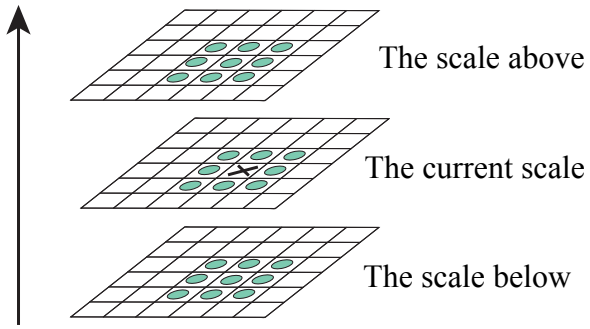



Figure 3.4: Maxima and minima of the DoG images are extracted by comparing a pixel (marked with $X$) to its eight neighbours in the current scale and nine neighbours in the scale above and below (marked with circles). Figure taken from Lowe (2004).

#### 3.2.2.2 2D SIFT Descriptor

The final step is to describe the keypoint based on the dominant orientation that achieves the rotation invariant. For each image scale $L(x, y, \sigma)$, the gradient magnitude $m(x, y)$ and orientation $\theta(x, y)$ are computed using pixel differences:

$$m(x, y, \sigma) = \sqrt{(L(x+1, y, \sigma) - L(x-1, y, \sigma)^2 + (L(x, y+1, \sigma) - L(x, y-1, \sigma)^2} \tag{3.5}$$

$$\theta(x, y) = tan^{-1}(\frac{L(x, y+1, \sigma) - L_(x, y1-1, \sigma)}{L(x+1, y, \sigma) - L(x-1, y, \sigma)}) \tag{3.6}$$

A neighbourhood region around each keypoint is then defined and the orientation of the gradient of the points in this region is represented by a histogram $H$ with 36 bins. The peak of $H$ is detected as the dominant orientation and assigned to the keypoint, where each keypoint has four values defining location $x$ and $y$, scale $\sigma$ and orientation $\theta$. For each keypoint a region around the point is considered and sampled to $4 \times 4$ sub-regions. Then a histogram with 8 bins is constructed to represent the orientation of each sub-region and concatenated to form a $(4 \times 4) \times 8 = 128$ descriptor vector (cf. Figure 3.5).

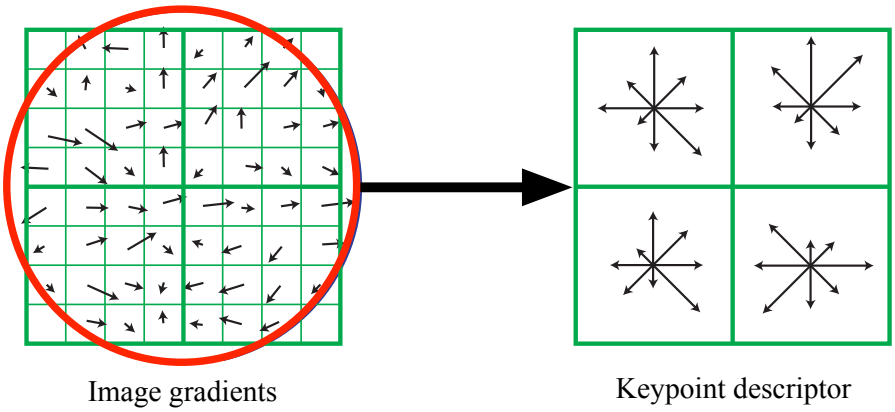



Figure 3.5: Constructing keypoint descriptor in 2D SIFT by firstly computing the gradient magnitude and orientation at each sub-region of a $4 \times 4$ subregion around the keypoint (shown on the left). These are then accumulated into orientation histograms of 8 bins (shown on the right) and constructed to form the descriptor vector. For illustration, the figure shows a $2 \times 2$ descriptor array rather than $4 \times 4$. Figure taken from Lowe (2004).

#### 3.2.2.3 3D SIFT Extensions

The previous extensions for the 2D SIFT can be categorised into three groups. The first one extended the descriptor part only and combined it with 2D detectors while the second provides a full 3D spatial extension for the 3D images. The last group combined different approaches to describe the motion and the appearance separately.

From the first category, Scovanner et al. (2007) extended the descriptor side to the time domain and dropped the scale and location invariance covered by the detector side. For each pixel, the gradient magnitude is computed as:

$$m_{3D}(x, y, t) = \sqrt{L_x^2 + L_y^2 + L_t^2} \tag{3.7}$$

where $L_x$, $L_y$ and $L_t$ are the pixel differences approximated by:

$$L_x = L(x + 1, y, t) - L(x - 1, y, t) \tag{3.8}$$

$$L_y = L(x, y + 1, t) - L(x, y - 1, t) \tag{3.9}$$

$$L_t = L(x, y, t + 1) - L(x, y, t - 1). \tag{3.10}$$

Each pixel is associated with two angle values to define the gradient direction as shown

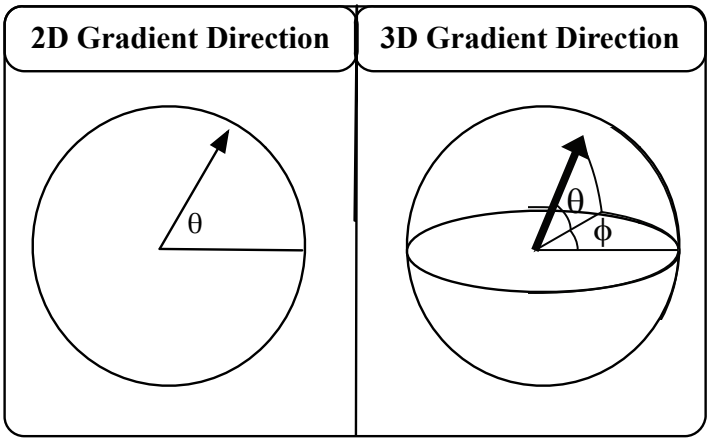



Figure 3.6: A visual representation for the angles in the 2D and the 3D gradient. Figure taken from Scovanner et al. (2007)

in Figure 3.6, where $\theta$ is the polar angle in the polar coordinates or 2D domain, and $\phi$ is the polar angle in the spherical coordinates or 3D domain:

$$\theta(x, y, t) = tan^{-1}\left(\frac{L_y}{L_x}\right) \tag{3.11}$$

$$\phi\,(x, y, t) = tan^{-1}\left(\frac{L_t}{\sqrt{L_x^2 + L_y^2}}\right) \tag{3.12}$$

The dominant orientation is determined by creating a 2D histogram with equally sized bins from the $\boldsymbol{\theta}$ values $\boldsymbol{\phi}$. Each 3D neighbourhood is rotated in the direction $\boldsymbol{\theta} = \boldsymbol{\phi} = 0$. This is accomplished by multiplying every pixel $(x, y, t)$ in the neighbourhood by the matrix

$$\begin{bmatrix} cos\theta cos\phi & -sin\theta & -cos\theta sin\phi \\ sin\theta cos\phi & cos\theta & -sin\theta sin\phi \\ sin\phi & 0 & cos\phi \end{bmatrix} \tag{3.13}$$

Finally, to create the descriptor, a factorisation of sub-histograms is computed for every $4 \times 4 \times 4$ sub-region with three values for each pixel: one magnitude and two orientations. Figure 3.7 illustrates the difference in the descriptor side between the 2D SIFT and the 3D SIFT.

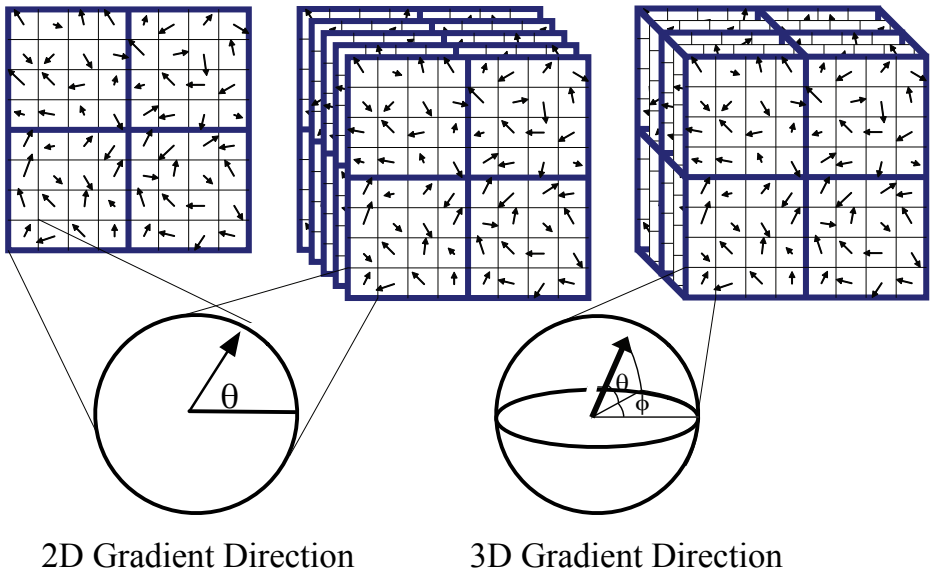



Figure 3.7: Comparison between the 2D and 3D SIFT descriptors. The left side shows the 2D SIFT descriptor. The centre shows the use of the multiple 2D SIFT descriptors in a video representation without modification. The right shows the 3D SIFT descriptor that defines 3D sub-volumes accumulated into a sub-histogram. The set of all histograms defines the final descriptor. Figure taken from Scovanner et al. (2007).

In the second category, Cheung and Hamarneh (2007) generalised the 128-D SIFT features to n-dimensional space (n-SIFT) using a $2^{5n-3}$-D features vector. Following the original SIFT algorithm by Lowe (2004), they build a multilevel pyramid of Gaussian image. The first level contains the Gaussian smoothed image at scale $\sigma$. Subsequently each level is a downsampled version of the previous one with sigma $= \sigma, K\sigma, K^2\sigma, \ldots, K^j\sigma$. At each scale $K^j\sigma$, the DoG is calculated between $K^j\sigma$ and $K^{j+1}\sigma$. At the DoG pyramid of nD-image, the local maxima is defined in the current scale by comparing the point value with the $3^{n+1} - 1$ neighbour points in the above scale ($K^{j+1}\sigma$) and the below scale ($K^{j-1}\sigma$). Finally, a threshold $T_{DoG}$ is used to filter the undesired local maxima, which are the extrema that have greater magnitude than $T_{DoG}$. To describe the features, they summarise $16^n$ regions around each feature where each region has $4^n$ subregions. Each voxel in the subregion is described by a histogram with $8^{n-1}$ bins, producing a feature vector with $2^{5n-3}$ dimensions.

Allaire et al. (2008) also provided a full 3D extension. They claimed that their

work addressed two important issues not solved in the previous extensions. The first problem is the poorly extracted points with low contrast and the other is the full 3D orientation invariant. Following the Lowe (2004) algorithm and similar to the work achieved by Cheung and Hamarneh (2007), the authors extended each step in features extraction and localisation to include the third parameter. The scale space $L(x, y, z, \sigma)$ is defined by convolution of the 3D image $I(x, y, z, \sigma)$ with Gaussian $G(x, y, z, \sigma)$ as:

$$L(x, y, z, \sigma) = G(x, y, z, \sigma) * I(x, y, z) \tag{3.14}$$

$$= \frac{1}{\left(\sqrt{2\pi\sigma}\right)^3} exp^{\frac{(x^2+y^2+z^2)}{2\sigma^2}} * I(x, y, z) \tag{3.15}$$

The DoG is computed to build a pyramid of blurred images and the local extrema is detected across 3D locations and scales as the features. Due to the nature of the medical images that produced a massive number of extrema with poor saliency, they applied a 3D filtering step that extended from the original 2D algorithm. To achieve the full orientation, they defined three angles for each feature rather than two as in Scovanner et al. (2007). Two angles: azimuth $Az \in [-\pi; \pi]$ and elevation $El \in \left[-\frac{\pi}{2}; \frac{\pi}{2}\right]$ were computed, similar to [Scovanner et al. (2007)] angles. The third one, $Ti \in [-\pi; \pi]$, is known as a roll or tilt angle and is built based on the value of the first two angles. In other words, the gradient magnitude and the 2D histogram for the angles is computed. The peak bin is used to generate the features. Then for each feature, another Gaussian histogram is created for the tilt angle, and the peak bin is used as the dominant orientation. For creating the descriptors, they created a 2,048-dimensional vector ($2^{5n-4}$) similar to [Scovanner et al. (2007)] using only the azimuth and elevation angles.

Unlike other groups, Chen and Hauptmann (2009) treated the spatial and temporal domains separately. Their motion SIFT (or MoSIFT) descriptor contains two parts, describing the spatial domain with HOG and the temporal domain with HOF, which capture the movement in the interest points. Firstly, a pair of frames is used to apply the normal 2D-SIFT algorithm and detect the distinctive interest points in appearance. Afterwards, the optical flow is utilised to filter those features with sufficient amounts of motion or action. Secondly, a pyramid of optical flow is constructed between each Gaussian pyramid from consecutive frames. Then local extrema are detected from the DoG pyramid if it contains motion information in the optical flow pyramid. In determining a descriptor, a factorisation histogram is used for each kind of feature

separately, with one difference in the dominant orientation. The optical flow does not involve orientation invariance. At the end, a single descriptor is created for both HOG and HOF features with 256 dimensions (128 SIFT + 128 optical flow).

## 3.3 Spatio-temporal SIFT Detector

As presented in Section 3.2 video representation task is still a challenging task and there is still a need for more development that interpret video content under various variations and environment changes. Most of the state-of-the-arts works related to spatio-temporal interest points detectors such as [Laptev and Lindeberg (2003),Dollar et al. (2005) and Schuldt et al. (2004)] either represent the appearance and motion information separately, sensitive to the significant variations in the spatial and temporal domains or produce small number of features which are unable to represent complex actions from reality.

To overcome the previous works limitations, this section presents the proposed ST-SIFT algorithm to detect and represent interest points in videos. Interest point detection converts a video signal from a 3D volume of pixels to a descriptive set of features [Lowe (2004)]. Ideally, these points should sample the portions of the video that contain activities while avoiding regions of low movement. The aim is therefore to deliver a technique that generates a sufficient but manageable number of features that can capture the information necessary to recognise arbitrary human activities. The proposed method identifies spatially distinctive regions that contain sufficient motion at a variety of spatio-temporal scales. The information in the neighborhood of each interest point is also considered using a 3D descriptor that explicitly encodes both appearance and motion components.

As presented in Section 3.2.2.3, there have been some previous works that extended the algorithm spatially to extract the extrema for the 3D images [Allaire et al. (2008); Chen and Hauptmann (2009); Cheung and Hamarneh (2007)] or to detect 2D interests points spatially and then described it with a 3D descriptor [Scovanner et al. (2007)]. Unlike these works, the defined ST-SIFT detects distinctive points from both space and time domains at multi-scales.

Figure 3.8 shows examples of interest points detected in an outdoor sequence from a KTH dataset with a hand-clapping woman and hand-waving man. The right side presents evidence that ST-SIFT interest points are much more meaningful and descriptive compared to those detected using the original SIFT detector on the right

side. The ST-SIFT approach tends to detect the points that correspond to the main participating body parts of the action being performed whilst those detected by the SIFT contain static body parts of high texture or background with strong edges.

The extended detector is more able to define video events under different transformations and circumstances. To achieve invariance in both space and time, firstly a spatio-temporal DoG pyramid from the Gaussian pyramid is calculated. Then three different planes $(xy, yt, xt)$ are used to extract the interest points from the DoG. The common points between those three planes comprise vital information in both spatial and temporal domains and are used to describe the video events. A pseudocode of the algorithm is provided in Algorithm 1.

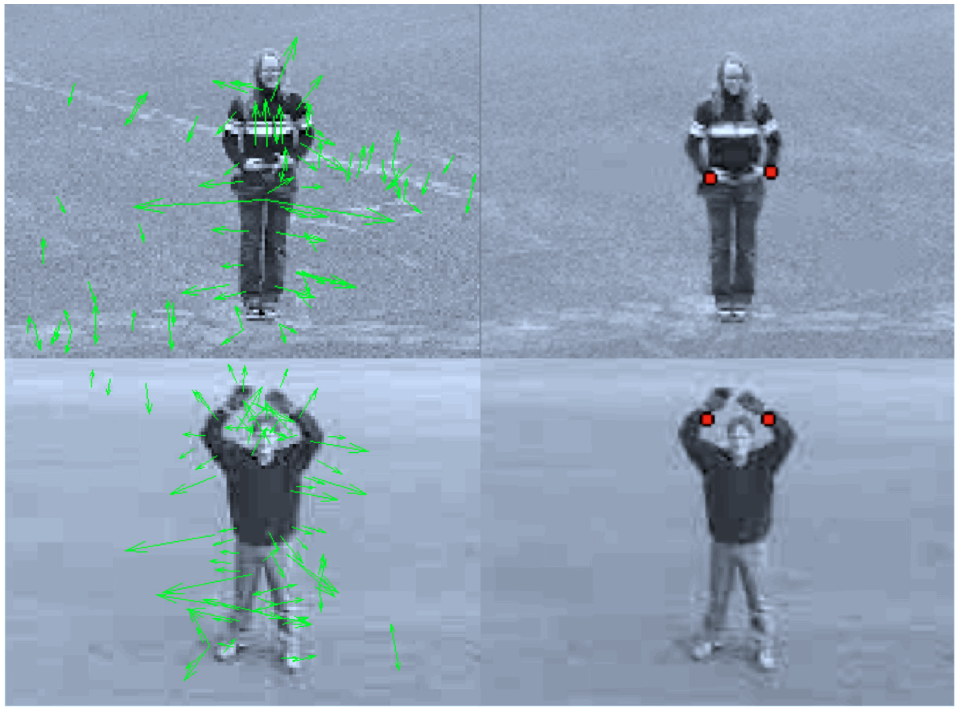


Figure 3.8: Sample frames from the KTH datasets showing hand clapping (top) and hand waving (bottom) actions. Interest points extracted with the proposed ST-SIFT (right) are compared with the 2D-SIFT (left). The 2D-SIFT defines the spatial points from both moving objects and the background, while the ST-SIFT defines the spatial points from different scales that only have motion information.

**Algorithm 1:** Scale-space extrema detection and Keypoint localization. Takes video stream as input and extracts interest points after computing the DoG pyramid. Parameters are number of levels, sigma for Gaussian filter and number of Gaussian scales at each level, which are pre-defined by experiment.

**input** : $F \rightarrow$ Set of Features , $S \rightarrow$ scale per level , $O \rightarrow$ number of level , $\sigma \rightarrow$ temporal scale , $\tau \rightarrow$ temporal scale , $t_c$, $t_e \rightarrow$ thresholds , $I = \{I_1, I_2, \ldots, I_N\} \rightarrow$ input image sequence , where $N = 2^{O+1}$

**output**: $F$

initialisation $L_1 = I$ , $F = \phi$ ;

**for** $i = 1$, *till* $O$ **do**

  $L_{i,1} = I_i$;

  **for** $j = 2$, *till* $S + 3$ **do**

    $L_{i,j}(.;\sigma,\tau) = L_i(x,y,t) * G(x,y,t,k^j\sigma,k^j\tau)$;

      where $G(x,y,t,k^j\sigma,k^j\tau) = \frac{1}{\sqrt{(2\pi)^3\sigma^4\tau^2}} exp\left(-\frac{\left(x^2+y^2\right)}{2\sigma^2} - \frac{t^2}{2\tau^2}\right)$ ;

      and $k = 2^{\frac{1}{S}}$ $\{See\ Section\ 3.3.1\}$ ;

    $DoG_{L_{i,j-1}} = L_{i,j} - L_{i,j-1}$;

  **end**

  $pI = Find\ Extrema\left(DoG_{L_{i,j-1}}, DoG_{L_{i,j}}, DoG_{L_{i,j+1}}\right)$ $\{See\ Section\ 3.3.2\}$ ;

  **forall the** $p \in pI$ **do**

    **if** $p > t_c$ **then**

      construct $3 \times 3$ Hessian matrix $H = \begin{bmatrix} D_{xx} & D_{yx} & D_{tx} \\ D_{xy} & D_{yy} & D_{ty} \\ D_{xt} & D_{yt} & D_{tt} \end{bmatrix}$ ;

        where $D_{ij}$ is the second derivative in the $DoG$ space ;

      compute $Trace(H) = D_{xx} + D_{yy} + D_{tt}$ ;

      compute

      $Det(H) = D_{xx}D_{yy}D_{tt} + 2D_{xy}D_{yt}D_{xt} - D_{xx}D_{yt}^2 - D_{yy}D_{xt}^2 - D_{tt}D_{xy}^2$ ;

      **if** $\frac{Trace^3(H)}{Det(H)} < \frac{(2t_e+1)^3}{(t_e)^2}$ **then**

        $F = F \cup$ Orientation assignment $(p)$

      **end**

    **end**

  **end**

  $L_{i+1} = Scale\left(G(x,y,t,\sigma), 0.5^i\right)$

**end**

### 3.3.1 Spatio-temporal Difference of Gaussian Pyramid

The DoG pyramid provides a bandpass version of the original signal [Dorr et al. (2010)], which serves as a scale-space of the video to help detect the invariant interest points. Unlike previous approaches in extending SIFT for constructing 3D spatial pyramids, this work treats both spatial and temporal domains equally. The spatio-temporal Gaussian pyramid was first introduced by Uz et al. (1991); in their study, downsampling was performed in both temporal and spatial domains simultaneously. This means every lower level in the pyramid is generated by dropping every other pixel in the spatial domain followed by dropping every other frame in the temporal domain. This process is adapted in this work to construct a multilevel spatio-temporal Gaussian and DoG pyramid as shown in Figure 3.9.

For a sequence of images $I$ of size $W$ (width) by $H$ (height) pixels, each pixel $(x, y, t)$ at location $(x, y)$ and frame $t$ is denoted as $I(t)(x, y)$. Similar to Lowe (2004), $S + 3$ scales for each of the $O$ levels in the Gaussian pyramid are generated to guarantee that the local extrema detection will cover a complete level. In the first step, the Gaussian pyramid is constructed with $O$ levels where $O$ is calculated from the frame size. Each level is referred to as $G_i(s)$, $0 \leq i \leq O$ and $0 \leq s \leq S + 3$, where the highest level $G_0$ is the original video signal.

Initially, the video signal $I$ is incrementally convolved with the 3D Gaussian filter to produce the scale space $L(x, y, t, \sigma, \tau)$ of the first level with multiple scales $S$ separated by constant $K = 2^{1/S}$ , *i.e.*

$$L\left(x, y, t, \sigma, \tau\right) = G\left(x, y, t, K\sigma, \tau\right) * I\left(x, y, t, \sigma, \tau\right), \tag{3.16}$$

where the spatio-temporal Gaussian function, with spatial scale parameter $\sigma$ and temporal scale parameter $\tau$, is defined by:

$$G\left(x, y, t, \sigma, \tau\right) = \frac{1}{\sqrt{\left(2\pi\right)^3 \sigma^4 \tau^2}} e^{\left(-\frac{\left(x^2+y^2\right)}{2\sigma^2} - \frac{t^2}{2\tau^2}\right)}. \tag{3.17}$$

Then each lower level is produced by spatially and temporally downsampling the Gaussian smoothed signal at scales $\sigma$ and $\tau$ in the previous level. This yields a level with a lower frame rate and frames that are half the size of the previous one (represented by the left side of each box in Figure 3.9). Therefore, given an original frame size of $W$ by $H$ pixels, a specific level $G_i(s)$ has $W/2^i$ by $H/2^i$ frame size and

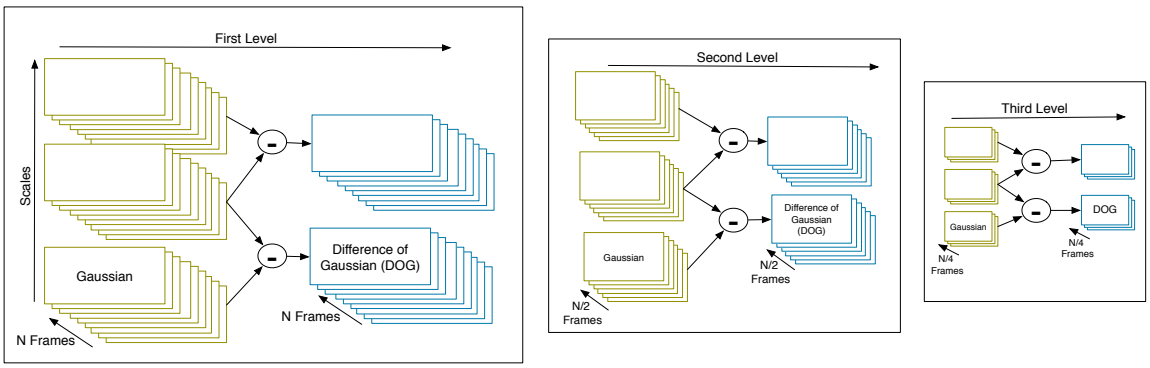



Figure 3.9: The video pyramids. Each level is a spatially and temporally downsampled version of the previous one, convolved with the 3D Gaussian to create the Gaussian pyramid. The DoG is then constructed by subtracting the adjacent Gaussian scales.

matches a point in the same time $t$ as $G_0(2^i s)$ ( as illustrated in Figure 3.10).

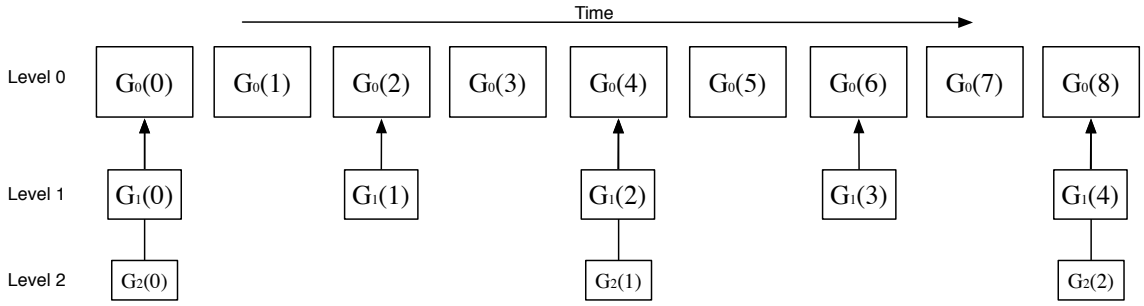



Figure 3.10: The mapping strategy between levels in the spatio-temporal Gaussian pyramid with three levels. Each pixel in the lower levels corresponds to a pixel in $G_0(2^i s)$, e.g. $G_1(3)$ is mapped to $G_0(6)$ and $G_2(1)$ is mapped to $G_0(4)$.

The second step constructs the DoG pyramid. For each level in the Gaussian pyramid, a DoG level is generated (with a number of scales is reduced by one) by subtracting the adjacent scales within the Gaussian level as shown in the right side of each box in Figure 3.9 and computed as following:

$$D(x, y, t, \sigma, \tau) = (G_K(x, y, t, K\sigma, K\tau) - G(x, y, t, \sigma, \tau)) * I(x, y, t, \sigma, \tau) \quad (3.18)$$

$$= L_K(x, y, t, K\sigma, K\tau) - L(x, y, t, \sigma, \tau) \quad (3.19)$$

### 3.3.2 Interest Points Detection

Once the DoG pyramid has been generated, the local extrema (minima/maxima) of the adjacent scales are combined from the $xy$, $xt$ and $yt$ planes. The assumption

here is that spatio-temporal events can be described by the common interest points between temporal (motion information) and spatial axes (appearance information).

Based on the work presented by Lopes et al. (2009), the video signal is a set of frames stacked together to form a spatio-temporal volume and there are two ways to slice this volume into frames (as illustrated in Figure 3.11). The first way is to slice through the spatial axis to create two types of spatial frames across the $\boldsymbol{x}$ or the $\boldsymbol{y}$ directions, while the second way creates the sequence of frames from the temporal axes combined with one of the $\boldsymbol{x}$ or $\boldsymbol{y}$ spatial axes. Therefore, three axis of each DoG scale volume represent $Space_X$, $Space_Y$ and time dimensions, with three faces represent ($Space_X$/$Space_Y$),($Space_X$/Time) and ($Space_Y$/Time). In this case, the extrema are detected from each face of spatio-temporal pyramid separately and the union of the results is taken. This allows the introduction of tolerance, *i.e.* it includes points where spatial and temporal extrema may not be at exactly the same pixel but are near to each other, and this tolerance is controlled by a threshold value.

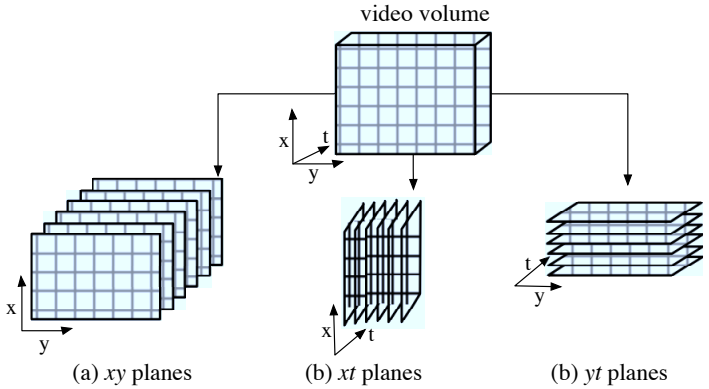



Figure 3.11: The video volume and the generated planes along different directions. Three axes of each DoG scale volume represent $x$, $y$ in the spacial and $t$ in the temporal dimensions, creating three spatio-temporal planes, $xy$, $xt$, and $yt$.

Similar to the 2D SIFT [Lowe (2004)], the local extrema are detected by comparing each sample point to its $3^{d+1}-1 = 80$ neighbours where $d$ is the three dimensions. The entire value is divided as 26 neighbours in the current scale (eight neighbours at the current frame, nine neighbours at the frame below and above) and then 27 neighbours in the scale above and below as shown in Figure 3.12. This is performed for each level within the DoG pyramid. At the end a filtering step is applied to remove the noises and edges' points.

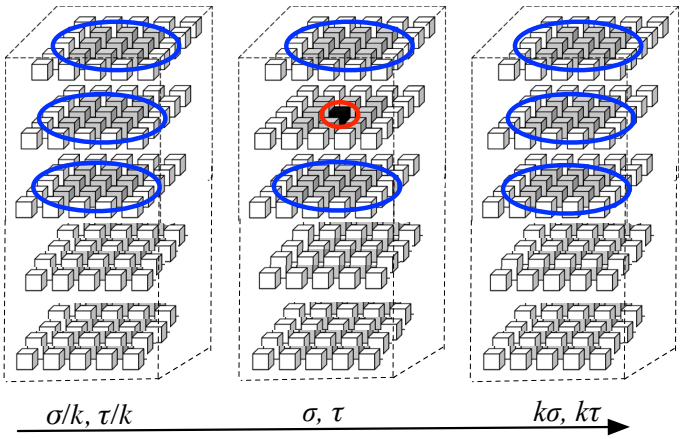



Figure 3.12: For video signal, an extrema (in black with red circle) is the maximum or minimum of the $3^{d+1} - 1 = 80$ neighbours (shaded with blue circle) where $d = 3$ is the dimension of the video. These neighbours are located in the same scale, the scale above and below.

## 3.4 Experiments and results

Action recognition is a hot research topic in computer vision and can be involved in various important applications such as video surveillance and the analysis of social interaction. Most of the spatio-temporal extensions and feature extraction methods have been evaluated by this task. To present the performance of the new extensions and prove its efficiency, this is the most suitable task to compare with previous works. Even though this task is out of this thesis scope, the proposed ST-SIFT detector was evaluated by classifying the presented activity in the given video. The aim was to define a rough position of ST-SIFT among the others and compare its performance with the state-of-the-art methods. The methodology we adopted is implemented in the VlFeat toolbox [Vedaldi and Fulkerson (2010)], which is an open library that contains various algorithms and applications for computer vision. One of the provided applications is an image classifier using 2D-SIFT features. The code was adapted to be used as an action classification framework in video data.

The first step is frame extraction using the FFmpeg[1], which helps to record, convert and stream multimedia data including audio and video into different formats. A collection of commands, free software and open source libraries are provided, such as libavcodec (the audio/video codec library), and libavformat (the input reading

[1]http://www.ffmpeg.org/

library). The default conversion rate is 25 fps; however, multiple frame rates were tested to be used in the implementation. The next step is to extract the interest points from the spatio-temporal video cube using the proposed detector. Subsequently, the spatio-temporal regions around the interest points are described using the 3D-HOG descriptors. The descriptor length is determined by the number of bins used to represent the $\boldsymbol{\theta}$ (the polar angle to define the interest point orientation in the polar coordinates or 2D domain) and $\boldsymbol{\phi}$ (the polar angle to define the interest point orientation in the spherical coordinates or 3D domain) angles in the sub-histograms. We used the publicly available code provided by Scovanner[2] with 640-dimensional descriptor, which was slightly different from what was described in Scovanner et al. (2007). They used a $2 \times 2 \times 2$ configuration of sub-histograms, and $N \times 4$ for the total number of pins that represents the gradient orientations, where $N$ is set to 20 and indicates the number of the mesh in a sphere defined for the orientation assignment step. As a result, the descriptor length has $(2 \times 2 \times 2 \times 80 = 640)$-dimensions.

A vocabulary learning step is then applied, where the generated descriptors from all the interest points are clustered to a pre-specified number of visual words. The k-means clustering method that implements the Elkan algorithm [Elkan (2003)] was used, as it is faster than the standard Lloyd k-means method [Lloyd (2006)]. The centres of the generated clusters are called 'visual words' while the set of these words is known as 'spatio-temporal word vocabulary'. Using the computed vocabulary, the frequency histogram where the visual descriptors for the videos are mapped to the visual words. The frequency of words in each video is then computed and accumulated into histograms that are known as signatures. Finally, these signatures are used by the SVM classifier to train representative models for each action. We use a non-linear SVM with $\chi^2$-kernel [Vedaldi and Zisserman (2012)].

### 3.4.1 Experimental Setup

From the set of publicly available human actions datasets introduced in Chapter 2 (Section 2.4), two were employed: the KTH dataset [Schuldt et al. (2004)] and UCF sports dataset [Rodriguez et al. (2008)]. The KTH is the standard dataset for this task and it serves as a consistent point of comparison against the current state-of-the-art methods. Since it is much smaller than other datasets and was taken over homogeneous backgrounds with a static camera, for robust comparison purposes another dataset

[2]http://www.vlfeat.org

with more movements and variations was required to evaluate the invariance of ST-SIFT to different changes. Therefore, the UCF sports dataset was included in the task: this contains realistic actions of different duration and scales with one or more persons. The actions are performed in different ways with a complicated background and large intra-class variations, which makes it a challenge for recognition and is suitable to verify the approach's features.

For the KTH dataset, tests were based on the studies conducted by Niebles et al. (2008) and Shao and Mattivi (2010); the dataset was divided into two parts, 16 persons for training and 9 persons for testing. On the UCF sports dataset, a leave-one-out cross-validation training method was used, following the original paper [Rodriguez et al. (2008)]. When constructing a Gaussian pyramid, the number of scales was set to three for each of four levels in the KTH dataset, and three for each of three levels in the UCF sports dataset. The codebook size was a key parameter for the vocabulary learning step. The experiment procedure by Shao and Mattivi (2010) was followed, and the best performance was obtained for both datasets with the codebook size of 1,500 words. A single SVM classifier was trained for classifying each action using all the samples except the subset left for the testing.

### 3.4.2 Results

This section presents the performance of the proposed detector using the two datasets. An evaluation of the results was carried out using a confusion matrix (or error matrix), which is a table layout that visualises the performance of a method based on false positives, false negatives, true positives, and true negatives [Stehman (1997)]. Each column of the table holds the instances of a predicted class, while each row holds the instances of an actual class. Its name comes from the fact that it aims to test whether the algorithm is confusing a pair of classes, *i.e.* mislabelling one as another. The most common statistic for calculating the performance from the confusion matrix is simple accuracy (acc), which is defined as the number of correct predictions across all classes divided by the number of classes.

Tables 3.1 and 3.2 show confusion matrices for human action classification experiments with the KTH and the UCF sports datasets respectively. The proposed method achieved an overall accuracy of 90.74% in the KTH dataset. Interestingly, "jogging" and "running" were confused with each other while the walking action was successfully recognised. It is reasonable to easily mis-classify "jogging" and "running" because these actions have a common motion patterns in the KTH dataset. Similarly, "hand-

| Actual Classes \ Predicted Classes | Walk | Jog | Run | Box | Wave | Clap |
|---|---|---|---|---|---|---|
| Walk | 100 | 0 | 0 | 0 | 0 | 0 |
| Jog | 0 | 78 | 11 | 0 | 0 | 11 |
| Run | 0 | 11 | 89 | 0 | 0 | 0 |
| Box | 0 | 0 | 0 | 100 | 0 | 0 |
| Wave | 0 | 0 | 0 | 0 | 100 | 0 |
| Clap | 0 | 0 | 11 | 0 | 11 | 78 |

Table 3.1: Confusion matrix for action recognition results with 90.74% recognition rate on standard KTH dataset containing six actions.

| Actual Classes \ Predicted Classes | Dive | Golf | Kick | Lift | Rid | Run | Skate | Swing | Walk |
|---|---|---|---|---|---|---|---|---|---|
| Dive | 75 | 25 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Golf | 0 | 75 | 25 | 0 | 0 | 0 | 0 | 0 | 0 |
| Kick | 0 | 0 | 100 | 0 | 0 | 0 | 0 | 0 | 0 |
| Lift | 0 | 0 | 0 | 100 | 0 | 0 | 0 | 0 | 0 |
| Rid | 0 | 0 | 0 | 25 | 50 | 25 | 0 | 0 | 0 |
| Run | 0 | 0 | 0 | 0 | 0 | 100 | 0 | 0 | 0 |
| Skate | 0 | 0 | 25 | 0 | 0 | 25 | 50 | 0 | 0 |
| Swing | 0 | 0 | 0 | 0 | 0 | 0 | 25 | 75 | 0 |
| Walk | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100 |

Table 3.2: Confusion matrix for action recognition results with 80.56% recognition rate on the UCF dataset that contains nine actions from sport categories.

clapping" was confused with "hand-waving" since these actions have similar motion patterns with non-moving objects and arms performing in a similar way.

For the UCF sports dataset, the proposed method successfully recognised most action categories and achieved promising results with an overall accuracy of 80.56%. Most of the mistakes were intuitively reasonable, e.g. "skateboarding" is confused with "running" and "kicking", "riding-hours" is confused with "running" and "lifting", "golf swing" is confused with "skateboarding", "diving" is confused with "golf swing" and "kicking". A possible explanation is that they share a similar spatio-temporal appearance and motion pattern. In addition, they were similar in the way that the moving objects passed on by the video.

In the second set of experiments, ST-SIFT was compared with two conventional SIFT algorithms. One was the combination of the original 2D-DoG detector and the

2D-HOG descriptor by Lowe (2004), in which each frame is represented separately. The other representation was the 2D-DoG detector and the 3D-HOG descriptor developed by Scovanner et al. (2007). This comparison aimed to show that the detector side plays an important role in boosting the recognition performance and that considering the spatio-temporal information only on the descriptor side is not enough to represent the performed action in video streams. Table 3.3 shows that the ST-SIFT detector followed by the 3D-HOG descriptor outperformed the other two representations. This indicates that ST-SIFT is able to (1) capture the interest points that have vital information in both the spatial and the temporal domains, which were missed by the conventional approaches, and to (2) represent events in real video sequences.

| detector | descriptor | KTH | UCF |
|---|---|---|---|
| ST SIFT | 3D-HOG | 90.74% | 80.56% |
| 2D-DoG | 3D-HOG | 77.00% | 77.78% |
| 2D-DoG | 2D-HOG | 72.22% | 58.52% |

Table 3.3: The performance of the proposed ST-SIFT on both KTH and UCF sport datasets compared with the original SIFT detector. Spatial interest points with 3D HOG are described in the second row, and with 2D-HOG in the third row.

In the KTH dataset, the presented work significantly boosted the results from the 2D-DoG detector and the 3D-HOG descriptor combination (by 13.74%). Since the ST-SIFT considered the appearance and motion in both detector and descriptor sides, it was more able to generate distinctive features with a high ability to classify most of the actions performed. Meanwhile the 2D-DoG detector combined with the 3D-HOG descriptor detected the spatial features with common appearance for most of the actions. This was not enough especially in the KTH dataset, where the actions are mostly captured between the frames. Their explicit representation of motion in the descriptor side was insufficient for human activity recognition.

On the UCF sports dataset, the 2D-DoG detector with the 3D-HOG descriptor achieved remarkably better accuracy (by 19.26%) than the 2D-DoG detector with the 2D-HOG descriptor. This is reasonable for the sports datasets that contain lots of people with irregular and rapid changes. Thus, motion captured by the 3D-HOG is more discriminative than static appearance captured by 2D-DoG.

### 3.4.3 Comparison of ST-SIFT with the State-of-the-Art

In this section, the performance of ST-SIFT is compared with approaches to interest points extraction using KTH and UCF sports data published recently. The purpose of this comparison is to show the rough position of ST-SIFT among the state-of-the-art techniques in the context of the action classification task. ST-SIFT was not the best but was comparable to the state-of-the-art techniques in the field. Action recognition is not the aim of this thesis; however, it is considered since it is one of the richest task with reported performances from various developed methods in video-processing. Note that most action recognition works either applied pre-processing steps such as object tracking or background subtraction before extracting interest points, or combined multiple techniques to improve the performance, or employed a post-processing steps such as clustering and fusion step. However, the ST-SIFT method still needs to be improved and prepared for many other tasks, which would make it more useful and beneficial. At this point, its ability to classify human actions from different datasets gives some confidence that it is ready to participate in solving the main problem in this thesis, *i.e.* video sequence alignment and similarity.

From the results presented in Table 3.4, the recognition rate of the proposed method for the KTH dataset was among the state-of-the-art results reported by various techniques for the action recognition task. Most of these works employed a BoF with different descriptors, captured the appearance and the motion separately by combining multiple approaches, applied pre-processing steps such as object localisation, performed a features learning step, *etc.*Instead, the ST-SIFT analyses information from both space and time and then detects the common features between them as action representation. For example, Schuldt et al. (2004) reported a 71.7% accuracy by combining Laptev and Lindeberg (2003) local space-time features with spatio-temporal jets to recognise complex motion patterns. Dollar et al. (2005) proposed sparse spatio-temporal features extended from the 2D corner detector and described the interest points region using the spatio-temporal cuboid and achieved a recognition accuracy of 81.2%. Laptev et al. (2008) improved the accuracy to 91.8% by describing interest points with the HoG, and the HoF followed the BoF approach. Niebles et al. (2008) on the other hand achieved a lower performance of 83.3% than Dollar et al. (2005) by describing the Dollar interest points with a spatial-temporal gradient cube. Kovashka and Grauman (2010) achieved the highest rate of 94.50% by applying Laptev et al. (2008) as interest points detector and then grouped the results into a hierarchical process to generate a set of features. Using a 3D R transformation method

to describe 3D Harris interest points, Yuan et al. (2013) achieved a good result of 87.33% . While, Le et al. (2011) improved the accuracy rate to 93.90% using an unsupervised deep learning algorithm to learn spatio-temporal features of interest points from unlabelled videos.

| Method | Detector | Descriptor | KTH |
|---|---|---|---|
| Yuan et al. (2013) | Harris3D | 3D R transform | 95.49% |
| Gilbert et al. (2009) | Dense corner features | Hierarchical group | 94.50% |
| Le et al. (2011) | Edge detector | ISA | 93.90% |
| Schindler and van Gool (2008) | Gabor filter | Optical flow | 92.70% |
| Laptev et al. (2008) | Harris3D detector | HOG/HOF | 91.80% |
| **Our approach** | ST-SIFT | 3D-HOG | **90.74%** |
| Nowozin et al. (2007) | Dollar | Dollar | 87.04% |
| Niebles et al. (2008) | Dollar | Gradient descriptor | 83.30% |
| Dollar et al. (2005) | Cuboid detector | Cuboid | 81.17% |
| Schuldt et al. (2004) | Harris3D detector | Spatio-temporal jets | 71.72% |
| Ke et al. (2005) | Extension of Viola & Jones | Integral video | 62.96% |

Table 3.4: Rough position of ST-SIFT among the state-of-the-art techniques for the human action classification task using the KTH dataset. Note that a strict comparison should not be made because experimental conditions may be different. The accuracies (recognition rates in %) are rounded to two decimal places.

For the UCF sports dataset as presented in Table 3.5, Rodriguez et al. (2008) achieved 69.2% accuracy by extending the traditional maximum average correlation height (MACH) filter to 3D volume and combining it with spatio-temporal regularity flow (SPREF). Liu et al. (2009a) reported a recognition rate of 74.5% by combining different detectors including Harris-Laplacian (HAR), Hessian-Laplacian (HES) and MSER; they described the region of interest using Dollar et al. (2005). Wang et al. (2009) outperformed them with 85.6% by sampling the video into set of blocks and applied the 3D-HOG descriptor. Kläser et al. (2010) boosted the recognition accuracy to 86.7% by employing object localisation in the BoF representation. A higher accuracy rate of 87.3% was reached by Kovashka and Grauman (2010) by learning the space-time neighbourhoods in a BoF representation. Yuan et al. (2013) were the best in the KTH dataset but not on the UCF sport dataset with accuracy rate of 87.33%. An even better result of 86.5% was achieved by Le et al. (2011) using an unsupervised deep learning algorithm.

Human actions, especially in the UCF sports dataset, contain camera motion that produces interest points that are not related to the performed activity. However, in most cases, ST-SIFT leads to more distinguishable interest points. Even with the small actions in the KTH that have common appearance information on a cluttered but sta-

| Method | Detector | Descriptor | UCF sports |
|---|---|---|---|
| Le et al. (2011) | Edge detector | ISA | 86.50% |
| Yuan et al. (2013) | Harris3D | 3D R transform | 87.33% |
| Kovashka and Grauman (2010) | Dense sampling | St-Harris | 87.27% |
| Kläser et al. (2010) | Harris3D detector | 3D-HOG | 86.70% |
| Wang et al. (2009) | Dense sampling | 3D-HOG | 85.60% |
| **Our approach** | ST-SIFT | 3D-HOG | **80.56%** |
| Yeffet and Wolf (2009) | Accumulated histograms | LBP | 79.20% |
| Liu et al. (2009a) | HAR, HES, MSER | Dollar | 74.50% |
| Rodriguez et al. (2008) | MACH filter | SPREF | 69.20% |

Table 3.5: Rough position of ST-SIFT using the UCF sports dataset, where accuracies (recognition rates in %) are rounded to two decimal places. Note that a strict comparison should not be made because experimental conditions may be different.

tionary background, the ST-SIFT was able to recognise the main action. Therefore, it is more able than the other reviewed detectors to represent video sequences with repetitive or similar contents. The focus in this chapter has been to extract the local features that are invariant to location, scale and orientation changes. Other applications that involve scaling and orientation changes would deliver a better performance for this type of features. However, the presented approach achieves comparable results with the state-of-the-art approaches in the field.

## 3.5 Conclusion

This chapter started the three-stage framework for video sequences alignment by developing the first stage which involves a video representation technique. A spatio-temporal extension to the 2D SIFT approach was introduced with an evaluation in the task of human action classification. The ST-SIFT detector was combined with the 3D-HOG descriptor and applied to the KTH and the UCF sports datasets. The results showed that ST-SIFT was able to detect local features for human activities.

The purpose of this development was to extract local features that were invariant to location, scale, orientation and temporal changes. The KTH and the UCF sports datasets involve few scaling and orientation changes to evaluate the invariance. however, ST-SIFT should be able to deliver good performance when such changes are significant. At this level, our method was able to build a comparison position among the existing approaches that are usually tested on these datasets for this task.

# Chapter 4

# Stage 2: Spatio-temporal Coding

The previous chapter presented the first stage of the video sequences alignment framework developed throughout this thesis. An ST-SIFT detector was defined to describe both appearance and motion of regions of interest at multiple scales from video streams. The aim was to reduce the video sequence from volume of pixels to a set of features extracted from the portions containing activities while avoiding regions of low movement. This chapter describes how the robustness of the video representation was improved by constructing a more informative and compact layer of representation using features coding techniques. A spatio-temporal coding technique is presented for a video sequence representation. The approach is based on the ST-SIFT features extraction technique combined with the locality-constrained linear coding (LLC) technique. The ST-SIFT is able to extract effectively the significant invariant local points in the spatial and temporal domains. The coding scheme is extended to project each spatio-temporal descriptor into a local coordinate representation produced by max pooling. The extension is evaluated using human action classification tasks. Experiments with the KTH, Weizmann, UCF sports and Hollywood datasets indicate that the approach is able to produce results comparable to those of state-of-the-art methods.

The chapter is structured as follows. Section 4.1 introduces the topic of this chapter combined with motivations and contributions of the presented work. Section 4.2 describes previously introduced approaches to features coding. One of these effective methods related to the developed work is also presented, and its extension to the spatio-temporal domain is proposed in Section 4.3. The performance of this extension is measured in the human action recognition task and compared to the conventional approaches as well as to the state-of-the-art methods in Section 4.4. A concluding discussion is set out in Section 4.5.

## 4.1 Introduction

Video representation is the fundamental problem in computer vision applications. Three types of features have been employed in the literature to represent visual content. Low-level features extracted directly from the pixels of the image such as color, texture, energy and pitch *etc.*, were used to describe individual components and provide details rather than an overview and have little or even no relation with the human perception [Smeaton et al. (2009)]. On the other hand, high-level features or semantic concept are more abstracted and used to describe the overall content, and typically focus on the image as a whole. They are usually defined by training a classifier using labelled data. Typical examples of this feature include objects such as 'chair', persons such as 'Barack Obama', scenes such as 'sky'. Between processing the low-level visual layer and analysing the high-level semantic layer, learning the mid-level layer was introduced [Duan et al. (2003)]. This is built upon low-level and typically close to image-level information without attempts to reach the high-level [Duan et al. (2003)], thus narrowing the gap between the syntax of the video stream captured by the low-level and the semantic meaning of that stream.

For video processing, a tremendous number of distinctive features are extracted. However, dealing with these features cannot achieve good retrieval results in most cases, especially when the high-level concepts defined by the user's interest is not easily represented using the low-level features. Instead of using this feature space, a hope was to find a hidden semantic 'concept' space that helps to bridge this gap between features and concept [Jiang et al. (2005)]. The dimensionality of this hidden space would be smaller than the original one, thus more representative, more readily processed and handled. To achieve this goal, most applications extracted low-level features and transformed them into a mid-level representations [Liu et al. (2008)]. Popular examples of mid-level features include BoW [Sivic et al. (2005)], spatial pyramids [Lazebnik et al. (2006)] and sparse coding (or SC) [Olshausen and Fieldt (1997); Yang et al. (2009)]. However, these methods have many problems, such as not considering spatial order of the features, which results in their failure to capture the location of the represented object. A number of techniques have been put forward to progress this area, including LLC [Wang et al. (2010)], which utilises the locality constraints to project the features into a local-coordinate system and then integrates them by pooling to generate a more compact representation. It proved its success in image processing applications; however, there is still more to achieve with respect to the

temporal information.

A key characteristic of a video sequence is its spatio-temporal information that delivers semantically coherent content [Singh et al. (2009)]. Chronologically ordered frames have objects with explicit spatial relationships and motion information inherited from their previous frames. Both temporal trajectories of spatial relations among objects and temporal trajectories of single objects to represent its activities are very important to define the video content. Therefore, considering only spatial information in the coding step is not sufficient for video sequence applications that contain all these complicated relationships. Unfortunately, temporal and spatial characteristics have not been sufficiently covered in the current coding techniques despite their obvious importance.

### 4.1.1 Motivations

In the past decades, various techniques, such as BoW and SC, have proved their effectiveness in image and video processing. The BoW method, which represents an image with a histogram of its local features, performs well at spatial translations of features, and image categorisation tasks. However, it disregards the spatial relationship between local features, which affect it is ability to capturing shapes or defining location of objects. As an improvement, the spatial pyramid matching (SPM) was proposed as coding technique and has shown promising results for image processing tasks. However, it needs to be combined with classifiers and non-linear kernels to perform well. Therefore, costly additional computational complexity is involved, which implies a poor scalability of this approach for video applications.

To fill this gap and solve the problem of the existing coding methods (*i.e.* losing relationship information and computational cost), LLC was developed as a faster and better representation for image features. However, all these works have been defined for image processing applications that consider only spatial information and ignore temporal information. Nevertheless there still a need in the video sequences application to define a mid-level representation that encodes the extracted features with fewer but more informative codes. This new level reduces the tremendous number of interest points usually extracted by features detectors and descriptors to a set of codes generated using a codebook defined from a sample of these features. Therefore, rather than considering each spatio-temporal descriptor as a separate feature, this step takes into account the fact that descriptor coding is an intermediate step in representing the whole video. Dealing with these codes as video representation

has computational advantages, saving both processing time in handling and space in storing visual descriptors.

### 4.1.2 Spatio-temporal Coding: Overview

This chapter presents a spatio-temporal extension of the LLC scheme for video classification tasks. LLC is an image processing coding scheme proposed by Wang et al. (2010) to encode a set of features extracted using 2D SIFT with a smaller (than the original set of features) set of codes based on the spatial relationship between the features. To detect interest feature points, the dense 2D SIFT is replaced, from the original work with ST-SIFT and the spatial graph, within the coding step, with a spatio-temporal shortest path graph. The ST-SIFT method is able to effectively extract the significant invariant points in the spatial and the temporal domains, while ST-LLC is able to represent these points with fewer codes using the spatio-temporal relationship between the descriptors and the basis codebook.

The approach consists of two principle stages. The first involves transformation of a 3D video signal into spatio-temporal pyramids, followed by extraction of the interest features using a ST-SIFT detector. The regions around these features are then described using the 3D-HOG descriptor developed by Scovanner et al. (2007). In the second stage, the LLC is applied to the extracted descriptor in order to encode the local descriptors with similar basis from a codebook. The approach is evaluated using a human action classification task as a benchmark. Even though action recognition task is not the focus of this thesis, it was chosen to evaluate the proposed approach as it is one of the richest tasks, with reported performances of new developed methods in video processing.

### 4.1.3 Spatio-temporal Coding: Contributions

The contributions of this work can be summarised as follows:

- Extension the LLC, which is an effective coding technique in the image domain, to the spatio-temporal video signal domain;
- Provision of a robust schema to represent human actions in video stream, which can be taken as a baseline for the feature research;
- Application of the spatio-temporal LLC for human action classification, achieving the state-of-the-art performance on several benchmarks.

## 4.2 Related Work

Use of mid-level features is a popular approach when transforming local image descriptors into image representations, as it can be easily used in matching and classification. Feature learning algorithms have been widely used in different tasks to convert the low-level features to mid-level features with richer representation. Significant improvements have been introduced over recent years but generally they can be summarised by the following steps. [Koniusz et al. (2013)]:

1. The regions of interest points are extracted using one of the descriptor approaches as image representation. A dictionary or visual vocabulary is then built, using a clustering technique such as k-means. These clusters are often called visual words, centres, atoms, or anchors.

2. A feature coding algorithm is then applied to embed the local descriptors into the visual vocabulary space. This generates mid-level features that express each descriptor by a set of visual words.

3. A pooling step is lastly employed to aggregate every local descriptor represented by a set of visual words into a single vector that represents the final signature.

Each step plays a vital role on the quality of the final image representation and thus affects the classification performance and computational speed [Koniusz et al. (2013)]. Various mid-level coding methods have been proposed to date in a number of benchmarks including BoW models [Csurka et al. (2004)] ( including hard assignment, soft assignment and localised soft assignment), the family of linear coordinate coding, entailing vector quantisation (or VQ) [Gray (1984); Lazebnik et al. (2006)], SC [Yang et al. (2009)] and LLC [Wang et al. (2010)].

Table 4.1 presents a comparison of these methods based on how they coded the visual appearance, considered the spatial coherence or locality, the quantisation error while they assigned features to the codes, their consistency in encoding similar descriptors and finally the computational cost of the process.

Mathematically, for all the reviewed methods, a matrix of input signal $X = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N] \in \mathbb{R}^{M \times N}$ is defined, where each column $\boldsymbol{x}_n$ is an $M$-dimensional column vector of extracted features such as SIFT. A collection of $K$ codewords is known as a dictionary or codebook where $V = \{\boldsymbol{v}_i : i = 1, 2, \ldots, K\}$ and $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_N]^T$ are the sets of the coefficients or weights associated with the visual words.

**Stage 2: Spatio-temporal Coding**

| Method | Appearance coding | Spatial locality | Quantisation error | Nonconsistent coding | Computational cost |
|---|---|---|---|---|---|
| BoW | Coding | | High | Low | Low |
| VQ | Coding | Pooling | High | Low | Low |
| SC | Sparse coding | Pooling | Low | High | High |
| LLC | Local coding | Pooling | Low | Low | Low |

Table 4.1: A comparison between features coding methods focusing on how they perform with respect to appearance, spatial locality, quantisation error through the coding process, non-consistent coding and computational cost. The methods are ordered starting from the BoW as the earliest method and ending with the more recent ones, including LLC; the methods were developed based upon their predecessor in an attempt to prevent shortcomings and solve the problems. Table is taken from [Xiao et al. (2014)].

### 4.2.1 Bag-of-Words model

The BoW model [Csurka et al. (2004)] was one of the first implementations in object retrieval, scene matching and visual categorisation. The local features in the training phase are clustered to define a codebook. Then the representation is generated by coding the extracted local features with the visual words from the pre-learned codebook. Various techniques have been proposed for the BoW coding, including hard assignment, soft assignment and localised soft assignment coding (LSC).

#### 4.2.1.1 Hard assignment coding

In the hard assignment coding [Lazebnik et al. (2006)], the basis of each descriptor is defined by assigning feature $x_i$ to its nearest word in the codebook $V$ using a specific distance metric. For example, if the Euclidean distance is used, then coding is defined as:

$$\alpha_{i,j} = \begin{cases} 1 & if\ j = \underset{j=1,\dots,K}{argmin} \|\boldsymbol{x}_i - \boldsymbol{v}_j\|_2^2 \\ 0 & otherwise, \end{cases} \tag{4.1}$$

where $\alpha_{i,j}$ is the code associated to the descriptor $\boldsymbol{x}_i$, and $\|.\|_2$ is the $l_2$-norm, and $K$ is the nearest neighbours.

#### 4.2.1.2 Soft assignment coding

In the soft assignment coding [van Gemert et al. (2008)], basis $\alpha_{i,j}$ is defined as the degree of membership of a descriptor $\boldsymbol{x}_i$ to the $j$th codeword $\boldsymbol{v}_j$.

$$\alpha_{i,j} = \frac{exp\left(-\beta \left\|\boldsymbol{x}_i - \boldsymbol{v}_j\right\|_2^2\right)}{\sum_{l=1}^{K} exp\left(-\beta \left\|\boldsymbol{x}_i - \boldsymbol{v}_l\right\|_2^2\right)} \tag{4.2}$$

where $\beta$ is the parameter used to controlling the softness of the assignment, and $\|.\|_2$ is the $l_2$-norm.

#### 4.2.1.3 Localised soft assignment coding

Liu et al. (2011) proposed LSC to improve the soft assignment coding by combining it with the localisation concept. The activation function in Eq 4.2 is redefined as follows:

$$\alpha_{i,j} = \frac{exp\left(-\beta d\left(\boldsymbol{x}_i, \boldsymbol{v}_j\right)\right)}{\sum_{l=1}^{K} exp\left(-\beta d\left(\boldsymbol{x}_i, \boldsymbol{v}_l\right)\right)} \tag{4.3}$$

$$d\left(\boldsymbol{x}_i, \boldsymbol{v}_j\right) = \begin{cases} dist\left(\boldsymbol{x}_i, \boldsymbol{v}_j\right) & if\ \boldsymbol{v}_j \in N_k\left(\boldsymbol{x}_i\right) \\ \infty & otherwise \end{cases} \tag{4.4}$$

where $dist\left(\boldsymbol{x}_i, \boldsymbol{v}_j\right)$ is the Euclidean distance between $\boldsymbol{x}_i$ and $\boldsymbol{v}_j$, $N_k$ is the set contains kNN of descriptor $\boldsymbol{x}_i$ defined by the distance $d\left(\boldsymbol{x}_i, \boldsymbol{v}_j\right)$.

### 4.2.2 Linear Coordinate Coding Techniques

In visual recognition researches, VQ has presented successful results. However there have been recent efforts to employ more powerful techniques for creating a high-level image representation [Lazebnik et al. (2006)]. SC has been considered as a good alternative to VQ and has produced excellent results in various recognition and classification tasks [Yang et al. (2009)]. To consider the locality of the feature space and to capture the information about the spatial layout of features, LLC was then presented, achieving good performance on several benchmarks [Wang et al. (2010)].

These methods are considered feature learning algorithms that contain two common steps: the training step and the encoding step. During training, basis functions should be learned as weights, codebook or dictionary. While in the encoding phase,

the learned values are used to map each input vector to the corresponding feature vector in the output.

#### 4.2.2.1 Vector Quantisation

Vector quantisation is a lossy compression technique that solves the following problem [Yang et al. (2009)]:

$$\min_{V} \sum_{n=1}^{N} \min_{k=1...K} \|\boldsymbol{x}_n - \boldsymbol{v}_k\|_2^2 \tag{4.5}$$

where $\boldsymbol{x}_n$ is the $n^{th}$ input vector, $\boldsymbol{v}_k$ is the $k^{th}$ element or codeword of the codebook $V$, and $\|.\|_2$ is the $l_2$-norm. Following the matrix factorisation concept, this problem can be solved as:

$$\min_{\boldsymbol{\alpha},V} \sum_{n=1}^{N} \|\boldsymbol{x}_n - V\boldsymbol{\alpha}_n\|_2^2 \tag{4.6}$$

$$\text{subject to } Card(\boldsymbol{\alpha}_n) = 1,\ |\boldsymbol{\alpha}_n| = 1,\ \boldsymbol{\alpha}_n \geq 0,\ \forall n$$

where $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_N]$ are coefficients or weights associated with each codebook, $Card(\boldsymbol{\alpha}_n) = 1$ is the cardinality constraint to have only one nonzero element in each $\boldsymbol{\alpha}_n$, $\boldsymbol{\alpha}_n \geq 0$ is the non-negative constraint (that all values of $\boldsymbol{\alpha}_n$ are non-negative), and $|\boldsymbol{\alpha}_n| = 1$ is the $L1$-norm of $\boldsymbol{\alpha}_n$. The optimisation in this equation involves two steps: the training step considers both $\boldsymbol{\alpha}$ and $V$, and the coding step where the updated $V$ is used to solve the problem with respect to the $\boldsymbol{\alpha}$. The result from this optimisation is the nonzero value in the $\boldsymbol{\alpha}_n$ used to determine the cluster of the $\boldsymbol{x}_n$ vector.

#### 4.2.2.2 Sparse Coding

Sparse coding is a set of algorithms for representing a signal as a linear combination of the basis functions [Olshausen and Fieldt (1997)]. These functions capture high-level features in the input data and reduce higher-order redundancy. Compared with other unsupervised methods for learning such as principal component analysis (PCA), SC generates a model of un-orthogonal basis functions that can be adapted easily. Moreover SC is over-complete; the number of basis functions exceeds the input dimensions. This leads to higher chances of finding better sparse representation for an input signal.

Reducing the high-order redundancy in data involves using a probabilistic model to define the image architecture [Olshausen and Fieldt (1997)]. This is accomplished by linearly decomposing the image into a set of basis functions which can be modified to best represent the image as a set of statistically independent events.The prominent feature of these events is the sparsity, where an input image can be interpreted with a few basis functions from a large set.

Mathematically, to obtain a sparse representation, the following optimisation problem is solved:

$$\min_{\boldsymbol{\alpha}_n} \{\|\boldsymbol{x}_n - V\boldsymbol{\alpha}_n\|_2^2 + \lambda \|\boldsymbol{\alpha}_n\|_1\} \tag{4.7}$$

where $\|\boldsymbol{\alpha}_n\|_1$ is the $l_1$-norm of the weight or coefficients vector $\boldsymbol{\alpha}_n$ that represents the number of non-zero elements, and $\lambda$ is a regularisation parameter that maintains the reconstruction error with respect to the sparsity.

This optimisation is a convex problem that can be solved in two iterative steps: SC with fixed dictionary $V$ and updating the dictionary with fixed $\boldsymbol{\alpha}$ [Lee et al. (2007)]. The approach keeps one variable fixed while minimising the other.

SC can be derived from VQ by modifying some of the constraints [Yang et al. (2009)]. Firstly, relaxing the constraint $Card\,(\boldsymbol{\alpha}_n) = 1$ by applying the $l_1$-norm regularisation on $\boldsymbol{\alpha}$. This enforces the $\boldsymbol{\alpha}_n$ values to be small non-negative elements. Secondly, not considering the constraint $\boldsymbol{\alpha}_n \geq 0$, because the sign of the $\boldsymbol{\alpha}_n$ is not important and it can be absorbed by letting $V^T \leftarrow \left[V^T, -V^T\right]$ and $\boldsymbol{\alpha}_n^T \leftarrow \left[\boldsymbol{\alpha}_{n+}^T, -\boldsymbol{\alpha}_{n-}^T\right]$, where $\boldsymbol{\alpha}_{n+} = min(0, \boldsymbol{\alpha}_n)$ and $\boldsymbol{\alpha}_{n-} = max(0, \boldsymbol{\alpha}_n)$ and with that the constraint can be trivially satisfied.

SC has been widely used in image representation, machine learning and signal processing due to its attractive characteristics [Yang et al. (2009)]. Firstly, unlike VQ, SC uses fewer constraints leading to lower reconstruction error. Secondly, sparse representation can efficiently model the salient features of images. Thirdly, statistical research on images has proved the sparse nature of the images.

A common path in image classification and recognition starts with extracting low-level features and transforming them into a global representation [Liu et al. (2008)]. This representation is defined as a new mid-level feature that is created from the low-level features and reflects the image information but not at a high level. SPM was defined based on the concept of a BoW model. Unlike the BoW, SPM considers

the spatial order of the features that make the representation more descriptive. It creates a model by using a low-level feature extraction technique followed by a coding approach such as VQ and ends with pooling methods such as max pooling [Serre et al. (2005)].

Chiu and Chen (2007) extended the standard SPM by replacing the VQ, as the applied coding technique, with SC. The approach divided an image into different scales of 21 segments ( $2^h \times 2^h,\ h = 0, 1, 2$ ). The SIFT features were extracted from each segment. The SC was applied to transform the low-level features into a desirable sparse representation. After that, all the codes combined with each low-level feature were pooled from different locations and scaled to produce more robust features to local transformation [Boureau et al. (2010)]. The work presented by Boureau et al. (2010) showed excellence performance by SC compared with hard quantisation. It was tested with multiple image databases and showed high ability in the classification task. Similarly, Yang et al. (2009) developed an extension of the SPM by replacing VQ with SC. This was derived by relaxing the cardinality restriction constraint of VQ, so each descriptor could be encoded by multiple basis. Although SC had shown remarkable effectiveness in representing feature quantisation, it suffers from two limitations: (1) even if there was a simple variation in local features, the response of basis in the dictionary could be quite different, and (2) it eliminates the interdependence and relationships between local features, which adversely affects the image representation.

Despite the fact that a variety of algorithms have been developed for classification and recognition tasks, several problems remain; the notable one is that, in general, a feature extraction method tends to sparsity, with the implication of high complexity levels for the selection of the model and learning. In response to this, a number of approaches were proposed that involve combining features learning techniques, dimensionality reduction approaches and clustering methods. The robust sparse coding (RSC) scheme, for instance, was presented by Yang et al. (2011), where SC was considered as a sparsity-constrained robust regression problem. RSC improved the performance of the original SC and proved its effectiveness in handling facial occlusions.

#### 4.2.2.3 Locality-constrained linear coding (LLC)

LLC is a coding scheme proposed by Wang et al. (2010) to project individual descriptors onto their respective local-coordinate systems. Locality is more important than sparsity with LLC because, although locality implies sparsity, the reverse does not hold. The use of the locality constraint in favour of the sparsity constraint in LLC

has the potential for a number of helpful properties. They may include:

- **Better reconstruction:** VQ represents each descriptor with a single basis from the codebook (as shown in Figure 4.1(a)), thus failing to capture the relationships between the basis in the codebook. In contrast, as illustrated in Figure 4.1(c)), LLC employs multiple basis to represents each descriptor. This means that while in the former approach similar descriptors may have very different codes, in the latter correlations between descriptors can be captured and the basis are shared;
- **Locally smooth sparsity:** reconstruction error is reduced in LLC through the use of multiple basis, *i.e.* its explicit locality adaptor makes sure that patches with similarities have correspondingly similar codes. On the other hand, as shown in Figure 4.1(b), the over-completeness of the SC codebook may allow it to represent quite different basis with similar patches to achieve the sparsity, thus losing correlations between codes;
- **Analytical solution:** spare coding requires computationally demanding steps, whereas LLC can be performed faster in practice.

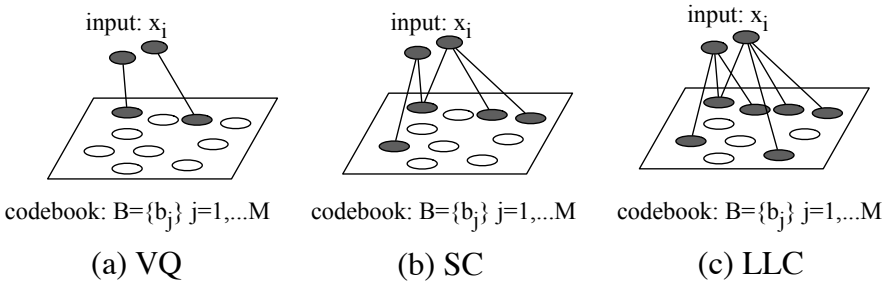



Figure 4.1: Comparison between the three linear coordinate coding techniques, (a)VQ, (b)SC, and (c)LLC in terms of assigning the codes to the set of features. The darker circles represent the selected bases or codes for the feature $\boldsymbol{x}_i$. Figure taken from [Yang et al. (2009)].

A codebook learning step is built into LLC via an 'online method' of learning [Wang et al. (2010)]; $B$ is the initial codebook that has been trained via k-means clustering. $B$ is then updated in increments as the training descriptors are iterated. For each of these increments, single or small-batch examples $\boldsymbol{x}_i$ are taken up and used to provide the required solution, resulting in the LLC codes associated with the current codebook $B$. This process takes the form of a feature selector, since it retains only a set of basis

$B_i$ with weights exceeding a predefined constant. The $\boldsymbol{x}_i$ values are then refitted but without locality constraints. The code then employed to update the basis using a gradient descent. Finally, the representational outcome (feature representation) is generated by submitting code to multi-scale spatial pyramid max pooling. There are clear benefits to this approach, for example speed, simplicity and scalability, while still providing comparable performance to the SPM [Yang et al. (2009)].

## 4.3 Spatio-temporal Coding

LLC performed remarkably better than the other coding techniques at different image-processing tasks. It is also considered simple, fast, scalable, and competitive. Since it only focuses on the spatial relationship between the extracted features and the codes, it cannot be performed effectively in video-processing applications. Considering the spatio-temporal information would provide a compact representation for video content and achieve robustness against different variations.

Most of the previous works reviewed in Section 4.2 were developed for the image-processing applications. These methods seek to reduce the high-order redundancy in the features and to generate more compact mid-level representation. Starting by the BoW, then extended it to the SC that considers the spatial order of the features, then extended it to the LLC that considers the locality of the spatial space. All these approaches performed remarkably good at different image-processing tasks and none of them consider the relationship between the features through the time domain. Therefore, adding the temporal relationship to these technique can be useful in generating more compact representation for video sequences.

This section fill in the gap in video content coding techniques by presenting a locality constrained spatio-temporal coding technique that considers the locality of the manifold structure in the input space. The notation used in the following sections is as follows:

- Let $X = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N] \in \mathbb{R}^{D \times N}$ be $D$-dimensional descriptors extracted from video stream using ST-SIFT as introduced in Chapter 3.
- Let $B = [\boldsymbol{b}_1, \ldots, \boldsymbol{b}_M] \in \mathbb{R}^{D \times M}$ be the codebook defined from the set $X$ using the clustering method.
- Let $S = [\boldsymbol{s}_1, \ldots, \boldsymbol{s}_N] \in \mathbb{R}^{M \times N}$ be the set of codes generated by the coding scheme to convert each spatio-temporal descriptor in $X$ into $M$-dimensional code as a

final representation.

The approach contains three steps; capturing video events with spatio-temporal local descriptors $X$, learning the locality-constrained sparse code $S$, and finally learning and optimising the codebook $B$ (see Figure 4.2).

### 4.3.1 Spatio-temporal interest points.

The set of interest points are firstly detected using the ST-SIFT detector. The video signal is firstly transformed into 3D spatio-temporal Gaussian and DoG pyramids. The local extrema are then extracted from the $xy$, $xt$ and $yt$ planes. Finally the regions around these detected points are described using 3D-HOG [Scovanner et al. (2007)], which calculates the spatio-temporal gradient for each pixel in the given cuboid. The approach leads to local regions that are invariant to scale and location in both the spatial and the temporal domains. More details are presented in Chapter 3.

### 4.3.2 Learning locality-constrained sparse coding.

LLC applies a locality constraint instead of the sparsity constraint used in the other coding techniques such as SC or RSC, as explained in Subsection 4.2.2. Following the original method by Wang et al. (2010), the criteria for the spatio-temporal LLC is defined as:

$$\begin{aligned} \min_{S} \sum_{i=1}^{N} \|\boldsymbol{x}_i - B\boldsymbol{s}_i\|^2 + \lambda \|\boldsymbol{d}_i \odot \boldsymbol{s}_i\|^2 \\ st. \quad 1^\top \boldsymbol{s}_i = 1, \forall i \end{aligned} \tag{4.8}$$

where $\odot$ is the element-wise multiplication, constraint $1^\top \boldsymbol{s}_i = 1$ is the shift-invariant requirement for the LLC code. that means the coding should not changed if the origin of the $RR^{D\times N}$ coordinate system for representing data points is changed, and $\lambda$ is a regularisation parameter that maintains the reconstruction error with respect to sparsity. The locality constrained parameter $\boldsymbol{d}_i$ represents every basis vector with different freedoms based on its similarity to the spatio-temporal descriptor $x_i$:

$$\begin{aligned} \boldsymbol{d}_i = \exp\left(\frac{dist(\boldsymbol{x}_i, B)}{\sigma}\right) \\ st. \quad dist(\boldsymbol{x}_i, B) = [dist(\boldsymbol{x}_i, \boldsymbol{b}_1), \ldots, dist(\boldsymbol{x}_i, \boldsymbol{b}_M)]^T \end{aligned} \tag{4.9}$$

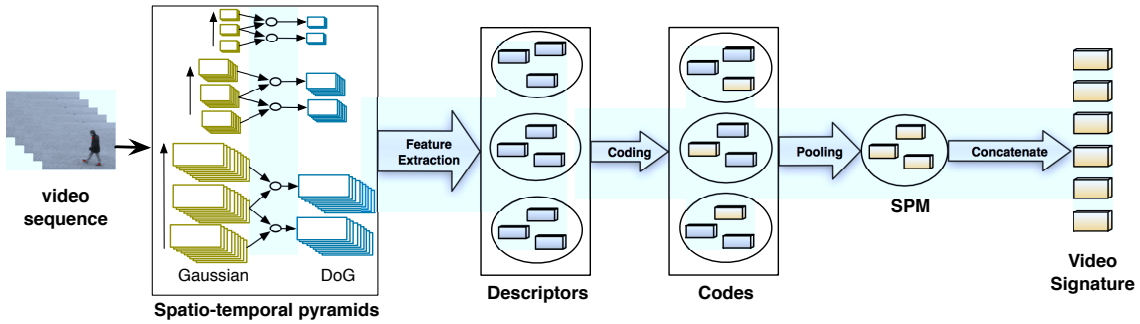



Figure 4.2: ST-SIFT is combined with LLC. The ST-LLC defines a mid-level of features containing a collection of codes as a video representation by extracting spatio-temporal interest points using ST-SIFT, learning the codebook and encoding features with these codes, pooling and finally concatenating them as a final representation.

where $dist(\boldsymbol{x}_i, b_1)$ is the geodesic distance between the spatio-temporal descriptor and the basis codebook, and $\sigma$ is the weight to control the locality parameter. The Euclidean distance used in the spatial LLC measures the distance between the descriptor and codes as the length of a straight line from one point to the other, whereas on the nonlinear manifold as video signal, the Euclidean distance between two points may not accurately reflect their intrinsic similarity. Using the geodesic distance helps to define the shortest path or curve that connects them in the high-dimensional space.

### 4.3.3 Codebook optimisation.

In Section 4.3.2, the codebook assumed to be given. The LLC used an on-line training method to train the codebook. Given a set of $D$-dimensional spatio-temporal descriptors, $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N$, an initial codebook is generated using the k-means clustering method. Optimisation is performed so that each spatio-temporal descriptor is approximated by the product of LLC coefficients and codebook. Using Eq 4.8, the optimal codebook can be constructed as [Wang et al. (2010)]:

$$\begin{aligned} \underset{S,B}{\operatorname{argmin}} \sum_{i=1}^{N} \|\boldsymbol{x}_i - B\boldsymbol{s}_i\|^2 + \lambda \|\boldsymbol{d}_i \odot \boldsymbol{s}_i\|^2 \\ st. \quad 1^\top \boldsymbol{s}_i = 1, \forall i \quad \|\boldsymbol{b}_j\|^2 \leq 1, \forall j \end{aligned} \tag{4.10}$$

This is a convex problem in $B$ only or $S$ but not in both together, and can be iteratively solved by the gradient descent method as following:

1. Initialise the dictionary $B$ with the codebook $B_{init}$ which is generated by clus-

tering such as K-means:

$$B \leftarrow B_{init} \tag{4.11}$$

2. For each spatio-temporal descriptor $\boldsymbol{x}_i$, compute the new LLC coefficient $\boldsymbol{s}_i$ using the current $B$ as in Eq 4.8.

3. From the set of basis $B$, remove the basis column $B_i$ with corresponding weights exceed a predefined threshold:

$$\begin{aligned} id &\leftarrow \{j \,|abs\,(\boldsymbol{s}_i\,(j)) > 0.01\,\} \\ B_i &\leftarrow B\,(:, id) \end{aligned} \tag{4.12}$$

4. Refit $\boldsymbol{x}_i$ without the locality constraint using an approximated code $\widetilde{\boldsymbol{s}}_i$; this step is to speed up the coding process.

$$\begin{aligned} \widetilde{\boldsymbol{s}}_i &\leftarrow argmax_s \left\|\boldsymbol{x}_i - B_i s\right\|^2 \\ s.t. \quad &\sum_j \boldsymbol{s}\,(j) = 1 \end{aligned} \tag{4.13}$$

5. Update the current dictionary:

$$\begin{aligned} \triangle B_i &\leftarrow -2\widetilde{\boldsymbol{s}}_i(x_i - B_i\widetilde{\boldsymbol{s}}_i), \\ \mu &\leftarrow \sqrt{\frac{1}{i}}, \\ B_i &\leftarrow B_i - \frac{\mu \triangle B_i}{|\widetilde{\boldsymbol{s}}_i|_2} \end{aligned} \tag{4.14}$$

6. Project those basis computed at each iteration onto the output matrix $B$:

$$B(:, id) \leftarrow proj(B_i) \tag{4.15}$$

## 4.4 Experiments and results

The human action classification task recently became a core interest domain as it is involved in a wide range of applications such as surveillance, human-machine interfaces and video indexing. Most of the new development in feature extraction or learning

domain would usually evaluated in solving action classification problem, giving us a wide chances to compare with. Classification of human action is not the focus on this thesis; however, this task was chosen for evaluation to estimate the position of ST-LLC among the other methods and compare its performance with state-of-the-art methods. It is also a challenging task due to the large variations in the time dimension combined with the usual difficulties in processing the frames; thus it gives confidence that the developed approach can effectively be applied to more advance tasks. The methodology adopted is implemented in the 2D-LLC based image classifier [Wang et al. (2010)], which contains three steps: feature extraction, descriptor coding and pooling to represent image content. The code was adapted to be used as action classification framework with video sequences.

For video representation, the same steps used in Chapter 3 were adopted in this experiment. Frames segmentation were firstly applied to generate sequence of frames. Spatio-temporal interest points were then detected using ST-SIFT. Finally, the spatio-temporal regions around the interest points were described using the 3D-HOG descriptors [Scovanner et al. (2007)]. In the SPM step, the ST-LLC codes were computed for each spatio-temporal sub-region and pooled together using multi-scale max pooling [Serre et al. (2005)] to create the corresponding pooled representation. The experiment used $4 \times 4$, $2 \times 2$ and $1 \times 1$ sub-regions. The pooled features were then concatenated and normalised using $\ell^2$-normalisation and used as the final representation for each action. Finally, SVM classifier was used to learn a model from signatures for each action. A non-linear SVM with a $\chi^2$-kernel was used. [Vedaldi and Zisserman (2012)].

The difference between this framework and the one used in Chapter 3 is the feature learning step. Chapter 3 applied the k-means clustering method to represent each action, followed by histogram representation that was used later to train the SVM classifier. This chapter, however, describes a process of applying the LLC with shortest path graph to generate the action representation, used later as input to the SVM classifier.

### 4.4.1 Experimental Setup

From the set of publicly available human actions datasets introduced in Chapter 2 (Section 2.4), four were employed: 1) the KTH dataset [Schuldt et al. (2004)], which is to date the most frequently used dataset employed to evaluate the human action task and many results have been reported on it. Thus it gives a good comparison to a wide range of published results in this task; 2) the Weizmann dataset [Blank

et al. (2005)], which has lower data scale than the KTH and contains only one scenario with static background. However, it consists of more action classes and higher inter-class similarity; 3) the Hollywood dataset [Laptev et al. (2008)] contains human activity collected from real-world movies, rather than laboratory collections as KTH and Weizmann. Since it is selected from movie scenes, it involves more cluttered backgrounds and the actions performed have more variety; and 4) the UCF sports dataset [Rodriguez et al. (2008)], a more challenging dataset with sequences mostly acquired by moving cameras. It contains a large intra-class variability in the action performed, camera motion, viewpoint, illumination and background.

Constructing a Gaussian pyramid as defined in Chapter 3 requires the definition of the number of levels with each containing a number of scales. The number of scales was set (by experiment) to three scales for each one of the four levels in the KTH and Weizmann datasets, and three scales for each one of the three levels in the UCF sports and Hollywood datasets. For the classification, the perform leave-one-out cross validation technique was used, where one SVM was trained for each class using all videos except one (chosen randomly) on which testing is to be performed. The performance result was then reported as the average on all of classes. The k-means method was applied to generate the initial codebook with size of 1024 words (the key parameter for dictionary training), the number of neighbours $K = 5$, $\lambda = 500$ in Equation ( 4.10) and $\sigma = 100$ in Equation ( 4.9).

### 4.4.2 Results

This section presents the performance of the proposed coding technique combined with the ST-SIFT as feature detector. The results were evaluated as in Chapter 3 Section 3.4.2 using a confusion matrix. For each dataset a table layout was constructed with each column holding instances of a predicted class, and each row holding the instances of an actual class [Stehman (1997)]. The content of the table can be defined as: true positives ($TP$) and false positives ($FP$) are the instances correctly and incorrectly predicted, whereas true negatives ($TN$) and false negatives ($FN$) are instances correctly and incorrectly not predicted. The overall accuracy is then defined as the number of correct predictions across all classes divided by the number of classes. The evaluation was based on two types of comparisons: first, by comparison with the original approaches to measure the effectiveness of the development on the performance; second, by comparison with state-of-the-art works published recently on the same task and using the same datasets.

For the first type of evaluation, ST-LLC was compared with two other combinations of methods. The first one measures the effect of the video representation phase provided by the ST-SIFT detector and 3D-HOG descriptor, on the performance of the coding phase covered by the modified version of the LLC. The second one combined the conventional 2D-LLC framework that contains a combination of the original 2D-DoG detector and the 2D-HOG descriptor which form the spatial SIFT developed by Lowe (2004), and the spatial LLC [Wang et al. (2010)]. The aim of this comparison is to show the effectiveness of considering the spatio-temporal information in firstly the video representation phase and secondly the coding phase, and that the developed approach achieved its goal in enhancing the original algorithm of LLC.

Table 4.2 shows that the proposed method outperformed the original LLC framework in action recognition using video sequences, achieving overall accuracies of 100% for KTH, 100% for Weizmann, 88.89% for UCF sports and 50% for Hollywood dataset.

In the four datasets, the modified coding phase boosted the results from the ST-SIFT detector, 3D-HOG descriptor and spatial LLC combination by 8.0% in the KTH dataset, 10.7 % in the Weizmann dataset, 5.3 % in the UCF dataset and 5.8% in the Hollywood dataset. Even though the video representation phase captures the appearance and motion in both space and time, the coding phase considers more the relationship between the descriptor and defines a new representation based on their similarity. Consequently, with the modified coding phase it was more able to distinguish between actions with high similarity in the environment and the actors than was with video representation phase.

Comparing with the spatial framework, the modified coding phase combined with the spatio-temporal video representation boosted the results by 50.3% in the KTH dataset, 52.0 % in the Weizmann dataset, 18.1 % in the UCF dataset and 30.6% in the Hollywood dataset. This was expected since the spatial framework only considers the spatial information in both video representation and coding. The human actions datasets contain similar movements with minor changes in the spatial information; thus ignoring the temporal information in these types of data would classify most of the unrelated videos as same action class. In such cases the spatial information is not enough and the features have to be captured spatially and tracked temporally and then coded into final representation based on their similarity

Note that considering the spatio-temporal information on the video representation phase also boosted the performance of the spatial framework, by 42.3% in the KTH dataset, 41.3 % in the Weizmann dataset, 12.8 % in the UCF dataset and 24.8% in

| detector | descriptor | coding | KTH | Weizmann | UCF | Hollywood |
|---|---|---|---|---|---|---|
| ST-SIFT | 3D-HOG | St-LLC | 100% | 100% | 88.9% | 50.0% |
| ST-SIFT | 3D-HOG | LLC | 92.0% | 89.3% | 83.6% | 44.2 % |
| 2D-DoG | 2D-HOG | LLC | 49.7% | 48.0% | 70.8% | 19.4% |

Table 4.2: The performance of the proposed St-LLC on four datasets compared with the original LLC. The first row contains the results of the proposed method, which is a combination of ST-SIFT detector, 3D-HOG descriptor and the modified LLC. The second row shows the results of the combined the ST-SIFT detector, 3D-HOG descriptor with the spatial LLC. The last row contains results for the original LLC framework with spatial approaches for image processing.

the Hollywood dataset. This is reasonable since the spatio-temporal video representation considered the appearance and motion in both detector and descriptor sides, thus generating more distinctive features with high ability to classify the actions performed, while the spatial framework extracted only the spatial features with common appearance in both objects and background for most of the actions.

### 4.4.3 Comparison with Recent State-of-the-Art

The performance of ST-LLC was compared with approaches published recently, and reviewed in Chapter 3 Section 3.2.1, in the human action recognition task using the four datasets. The aim was to test the performance of the video representation at this point and have the confidence that it is ready to be applied for the task related to video sequences alignment. The results achieved by ST-LLC were not the best but are roughly comparable to the current state-of-the-art. Note that results are often non-comparable due to the different experimental settings used in each approach.

From the results presented in Table 4.3, recognition rates for the proposed method in action recognition task for the KTH and Weizmann datasets exceed the state-of-the-art results reported by various techniques, and for UCF sports and Hollywood datasets were not the best but were comparable to the state-of-the-art results. The KTH and the Weizmann data are technically 'solved' datasets, as the classification accuracy of 100% was reported by several groups recently. Sun et al. (2011) reached 100% for both datasets, while Weinland et al. (2010), Schindler and van Gool (2008), Yuan et al. (2013) and Yeffet and Wolf (2009) achieved 100% for the Weizmann data. Previously Yao et al. (2010) performed at 97.8% with the Weizmann dataset using the Hough transform voting framework. Campos et al. (2011) achieved an accuracy of 96.7% with the Weizmann dataset and 93.5% with the KTH dataset by applying BoW

| | KTH | Weizmann | UCF | Hollywood |
|---|---|---|---|---|
| —this work— | 100% | 100% | 88.9% | 50% |
| Yuan et al. (2013) | 94.4% | 100% | 91.3% | 44.5% |
| Sun et al. (2011) | 100% | 100% | 86.9% | – |
| Chen and Hauptmann (2009) | 95.8% | – | – | 30.9% |
| Gilbert et al. (2009) | 94.5% | – | – | 53.5% |
| Campos et al. (2011) | 93.5% | 96.7% | 80.0% | – |
| Schindler and van Gool (2008) | 92.7% | 100% | – | – |
| Weinland et al. (2010) | 92.0% | 100% | 87.7% | – |
| Klaser et al. (2008) | 91.4% | 84.3% | – | 24.7% |
| Yeffet and Wolf (2009) | 90.1% | 100% | 79.2% | 36.8% |
| Liu and Shah (2008) | 84.1% | – | – | – |
| Laptev et al. (2008) | 91.8% | – | – | 27.0% |
| Wu et al. (2011) | 95.7% | 92.8% | 89.7% | 47.6% |
| Yao et al. (2010) | 93.5% | 97.8% | 86.6% | – |

Table 4.3: Rough position of ST-LLC using the KTH, Weizmann, UCF sports and Hollywood datasets, where accuracies (recognition rates in %) are rounded to one decimal point. Note that strict comparison should not be made because experimental conditions may be different.

and spatio-temporal shapes to represent human actions. For the KTH data, Chen and Hauptmann (2009) reported 95.8% and Wu et al. (2011) 95.7%. Gilbert et al. (2009) achieved 94.5% using a mined dense spatio-temporal features method. More, recently, Yuan et al. (2013) achieved 94.4% using a manifold learned from fusing of different features.

For the UCF sports data, the ST-LLC (88.9%) outperformed most of the recently reported results including 79.2% by Yeffet and Wolf (2009) and 80.0% by Campos et al. (2011). Wu et al. (2011) applied a method based on Lagrangian particle trajectories and boosted the accuracy to 89.7%. The ST-LLC is in line with the state-of-the-art results published recently by Sun et al. (2011) (86.9%), Weinland et al. (2010) (87.7%) and Yao et al. (2010) (86.6%).

Finally for the Hollywood dataset, to our knowledge, the current best result was produced by Gilbert et al. (2009) (53.5%) using the hierarchical data mining approach. Other reported results include Chen and Hauptmann (2009) (30.9%), Klaser et al. (2008) (24.7%), Laptev et al. (2008) (27.0%), Yeffet and Wolf (2009) (36.8%) and recently Yuan et al. (2013) (44.5%).

## 4.5 Conclusion

This chapter has presented the second stage of the three-stage framework developed for video sequences alignment In the previous chapter, the 3D video volume was mapped to a set of spatio-temporal descriptors. The extracted points are invariant to various changes and the method proved its efficiency in human action recognition task. However, applying local features extraction methods on video streams would generate tremendous numbers of interest points. A mid-level technique is therefore applied to code the defining descriptors into a global representation used for matching and classification.

In this chapter, a spatio-temporal coding technique was presented based on ST-SIFT descriptors for the human action recognition task. The method extended the LLC approach by utilising the ST-SIFT descriptor to densely extract salient feature points from a 3D signal. This produced a group of distinctive feature points which were invariant to scale, rotation and translation as well as being robust to temporal variation. LLC was then employed to represent these points with fewer codes using the spatio-temporal relationship between the features and a predefined codebook. The experimental results showed that LLC with the ST-SIFT outperformed (or at least achieved as much as) most of the state-of-the-art approaches on human action classification benchmarks, including the KTH, Weizmann, UCF sports and Hollywood datasets.

# Chapter 5

# Stage 3: Spatio-temporal Manifold Representation

Chapter 2 presented the three-stage framework for video sequences alignment, which was evaluated later with different applications in the video similarity area. Following that, Chapter 3 presented a spatio-temporal interest points detector that represents both appearance and motion of a region of interest at multiple scales from a video. Then a coding schema to define mid-level features, which give a more compact and richer representation, was defined in Chapter 4. The last stage of the framework seeks to map the high-level representation, developed in stages 1 and 2, to the low-dimensional space using a manifold embedding. The manifold instances are reordered and clustered, creating a group of similar contents in the context of spatio-temporal similarity.

This chapter presents a spatio-temporal manifold embedding to identify and align nearly-repetitive contents in a video stream. The similarities observed in frame sequences are captured by defining two types of correlation graphs: an intra-correlation graph in the spatial domain and an inter-correlation graph in the temporal domain. The presented work is novel in that it does not utilise any prior information such as the length and contents of the repetitive scenes. No template is required, and no learning process is involved in the approach. Instead, a spatio-temporal inter-correlation between repeated video contents is defined by extending the spatial Isomap to spatio-temporal graph-based manifold embedding (or STG-Isomap) that captures the similarities between repetitive sequences. Firstly spatio-temporal features are defined for a high-level semantic representation of complex scenes in a video sequence. Interest points that have significant local variations in both space and time are extracted and

encoded using fewer codebook bases in the high-dimensional feature space. The ST-LLC was used as it is able to detect features using ST-SIFT. At each time instance (a video frame, practically) visual features, defined by the ST-LLC codes, are modelled to form a temporal coherence to adjacent frames. Secondly, the similarity is measured by constructing a shortest-path graph with k-nearest neighbour (kNN) in both the spatial and the temporal domains. An extension of Isomap is introduced, aiming to identify the underlying structure in repetitive video sequences, semantically modelled by ST-LLC codes. The structure of heterogeneous data is reconstructed, presenting clusters of repetitive scenes. Experiments using a TRECVID rushes video proved that the framework was able to improve the embedding of repetitive sequences over the conventional methods, and thus was able to identify the repetitive contents from complex scenes.

The chapter is structured as follows. Section 5.1 introduces the topic of this chapter combined with motivations and contributions of the presented work. Section 5.2 describes previously developed approaches to manifold embedding and dimensionality reduction. One of these effective methods related to the developed work is also presented, then its extension to the spatio-temporal domain is proposed in Section 5.3. The performance of this extension in identifying and aligning nearly-repetitive contents is measured and compared to the conventional approaches in Section 5.4. A concluding discussion is set out in Section 5.5.

## 5.1 Introduction

Manifold learning has been a vital tool for various applications in computer vision and pattern recognition. It is a class of nonlinear dimensionality reduction techniques that transfer data from a high-dimensional space to a suitable output space with reduced dimensionality [van der Maaten et al. (2008)]. Nonlinear manifold learning does not assume the linearity of the input space, thus providing a better chance of dealing with input data with complex embedding in the high-dimensional space. The goal is to map a set of high-dimensional data into a low-dimensional space, while retaining the intrinsic structure in the data. The space with reduced dimensions should reflect the intrinsic dimensionality of the data, that is, the least number of parameters that capture the data features. These techniques can be also considered visualisation tools, where the high-dimensional data set is projected into two or three dimensions for display, which gives a better insight of the problem to be resolved. One drawback

of these methods if applied to the space-time domain is the possibility of temporal information loss, which is so hard to be recovered.

Various problems have been addressed using manifold techniques. Because they involve time series data, one potential direction would be to exploit the temporal information in learning the reduced dimensionality space. As far as we are aware, in the manifold learning literature, only Jenkins and Matarić (2004) took temporal coherency explicitly into account. Their algorithm extended the spatial Isomap by assigning similar low-dimensional weights to temporally adjacent samples extracted using a windowing technique. They grouped these samples so that temporally adjacent groups would have similar low-dimensional coordinates. They did not model dynamics and their performance depends on the window size, where smaller windows produced better results.

In computer vision, dimensionality reduction has played significant roles in various problems such as data compression, visualisation and classification due to its ability in eliminating useless high-dimensional features [van der Maaten et al. (2008)]. The underlying assumption for manifold learning techniques is that for each data point, there is a specific structure in its high-dimensional space location that can be exploited and summarised by a small number of attributes. These techniques have become increasingly important in the control and management of large and complex data sets such as nearly-repetitive contents that require a huge number of features.

### 5.1.1 Motivations

Repetitive contents in multimedia are frequently found in a combination of textual, visual and audio (speech) information. A quick search for any multimedia material using conventional search engines often results in multiple items with similar, or even identical, contents in the highest rank. In news broadcasts, for example, we frequently see nearly-repeated video footage although the presentation may vary with, for example, camera settings and appearance of objects, reflecting production processes and policies. Repetitive contents are not copies, but there exist some differences, thus making their management a difficult problem.

The task of aligning video sequences with repetitive contents requires careful synchronisation in both the spatial and temporal domains. The majority of previous work employed extra techniques such as template matching, camera calibration analysis and object tracking, which need more computation time and space to store. Additionally, the data in the high-dimensional space is too sparse to be efficiently used in video pro-

cessing and analysis [Lin and Zha (2008)]. Most of the features in high-dimensional space are just noise and may affect the learning process for pattern detection. The difficulty therefore lies in processing datasets with large numbers of features and a small number of learning samples, problem named as the 'curse of dimensionality' [Donoho (2000)]. Using dimensionality reduction methods to reduce the number of features before the training process can circumvent this problem.

Among the family of manifold learning approaches is the Isomap, which aims for a lower-dimensional embedding that maintains geodesic distances between all points. It is a graph-based scheme based on the multidimensional scaling (MDS) [Borg and Groenen (2007)]. Isomap overcomes dimensionality reduction problems by considering the geodesic distance, which defines the shortest path or curve that connects two points in a connected set. Unlike other manifold learning approaches, Isomap depends on a specific distance measurement between data points, which makes it possible to extend (due to its foundation in multidimensional scaling) [Jenkins and Matarić (2004)]. On the other hand, Locally Linear Embedding (LLE) [Roweis and Saul (2001)] for instance is defined using weights and locally linear models, which are naturally spatial and hard to extend. While, Kernel principal component analysis (KPCA) [Schölkopf et al. (1998)] applied local kernels globally from each point, but may not be globally suitable. One drawback of Isomap method if applied to the space-time domain is the possibility of temporal information loss, which is hard to recover.

### 5.1.2 STG-Isomap: Overview

This chapter presents a spatio-temporal framework for aligning nearly-repetitive contents. Embedded repetitions in the 3D signal, consisting of two spatial and one temporal dimension, are discovered by defining the coherent structure. The proposed method departs from the previous extension made on Isomap [Chantamunee and Gotoh (2010)] to spatio-temporal graph-based manifold embedding in that they defined an intra- and inter-correlation within and between repetitive scenes, and they construct a graph-based representation based on the similarity between frames and then mapped it to the lower-dimensional space. Chantamunee and Gotoh (2010) modelled the repetitive sequences using the GMM (Gaussian mixture model) and then the similarities between the frames are estimated using MLE (maximum likelihood estimation). A kNN graph between these models is then constructed and projected to the lower-dimension using MDS. The similarity within the video contents was observed based on frame-by-frame processing, which is time-consuming and expensive to apply

to longer video sequences. Additionally, they segmented the video sequence into a set of repetitive content and then computed the similarity between them.

The proposed method defines a spatio-temporal inter-correlation between repeated video contents by extending the spatial Isomap to a spatio-temporal graph-based manifold embedding that captures the similarities between repetitive sequences. It is an unsupervised approach that does not require segmentation or utilise prior information such as the number, the length or content of repetitions. Firstly spatio-temporal features are defined for a high-level semantic representation of complex scenes in a video sequence. Interest points that have significant local variations in both space and time are extracted and encoded using fewer codebook bases in the high-dimensional feature space. An extended version of the LLC (cf. Chapter 4) was used, which is able to detect features using ST-SIFT (cf. Chapter 3). The ST-LLC encoded each spatio-temporal descriptor by kNN based on the geodesic distances, instead of the Euclidean distances. The latter measures the distance between two points as the length of a straight line from one point to the other; but on the nonlinear manifold, the Euclidean distance may not accurately reflect their intrinsic similarity, which is measured by the geodesic distance. At each time instance, the visual features defined by the ST-LLC codes are modelled to form a temporal coherence to adjacent frames. At this point, the intra-correlation is derived between the codes of the video sequence, which helps to define the relationship between the features extracted from the video content. Secondly, a manifold representation maps the video sequence modelled by ST-LLC codes to the embedded space. At this stage the inter-correlation is computed between multiple video scenes by constructing a shortest-path graph with kNN in both the spatial and the temporal domains. The structure of heterogeneous data is reconstructed, presenting clusters of repetitive scenes.

### 5.1.3 STG-Isomap: Contributions

The contributions of this study are as follows:

- A spatial intra-correlation representation is created for repetitive contents in a video stream. Spatio-temporal interest points are extracted and encoded using fewer codebook bases in the high-dimensional feature space. Intra-correlation is then derived by constructing a shortest path graph using the kNN with geodesic distances.

- Isomap is extended to estimate the underlying structure of repetitive contents

and to define a spatio-temporal inter-correlation in a video stream.

- An unsupervised framework for aligning similar contents, which does not require prior information or pre-processing steps, is presented for multimedia data with repetitions.

## 5.2 Related Work

Dimensionality reduction is a transformation of a high-dimensional dataset into a lower-dimensional space, while preserving the useful structure of the original dataset [Law (2006)]. The underlying assumption for dimensionality reduction techniques is that data points do not lie randomly in the high-dimensional space, rather, there is a useful structure in their high-dimensional space locations that can be exploited and summarised by a small number of attributes.

The problem on the dimensionality reduction can be summarised as follows. Assume a dataset $X \in \mathbb{R}^{N \times D}$ with dimensionality $D$, and intrinsic dimensionality $d$ (where $d < D$, and often $d \ll D$). Intrinsic dimensionality is defined as the points in dataset $X$ that lie on or near a $d$-dimensional manifold that is embedded in the $D$-dimensional space. Dimensionality reduction techniques transform dataset $X$ with dimensionality D into a new dataset $Y$ with a smaller dimensionality $d$, while preserving the geometry of the data.

Dimensionality reduction techniques can be broadly divided into linear and nonlinear techniques depending on the nature of input data. Linear techniques depend on the assumption that the input data lie on or near a linear sub-space of the high-dimensional space. Nonlinear techniques do not make this assumption, and as a result can be applied to data in the high-dimensional space with more complex embeddings. Figure 5.1 shows a taxonomy of dimensionality reduction techniques. The further subdivisions in the taxonomy are discussed in the following three sections.

### 5.2.1 Preliminary

Throughout this section, we denote a high-dimensional data set by $X \in \mathbb{R}^{N \times D}$ consisting of $N$ data vectors $x_i\,(i \in \{1, 2, \ldots, N\})$ with dimensionality $D$. The lower-dimensional point for $x_i$ is defined by $y_i$ from the data set $Y \in \mathbb{R}^{N \times d}$ with intrinsic dimension $d$ (where $d < D$, and often $d \ll D$). The aim of the dimensionality reduction methods is to find transformations of $X$ that maps $x_i$ to to its low-dimensional

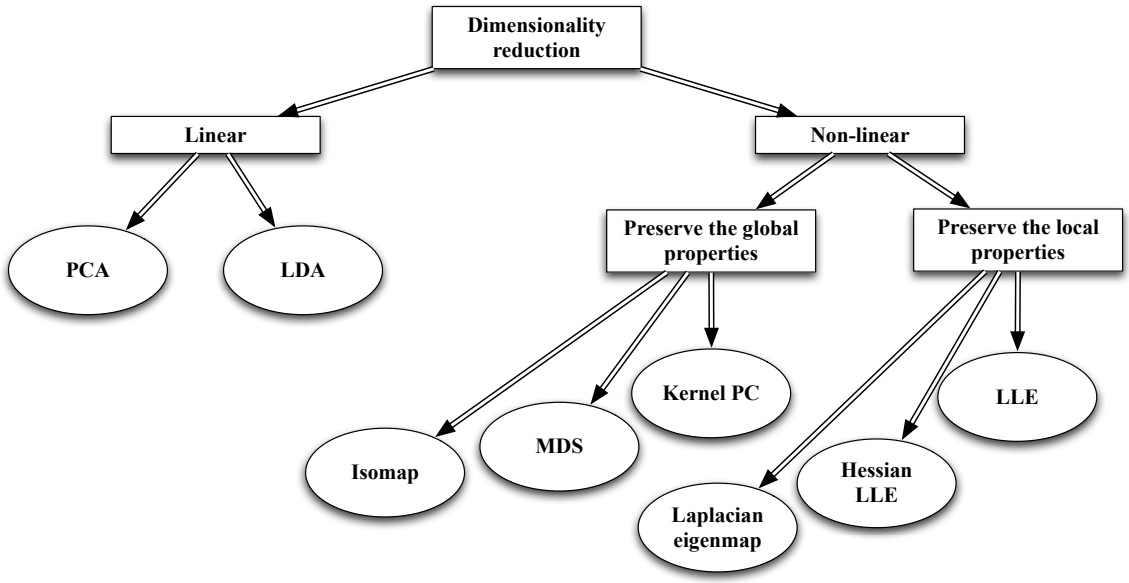


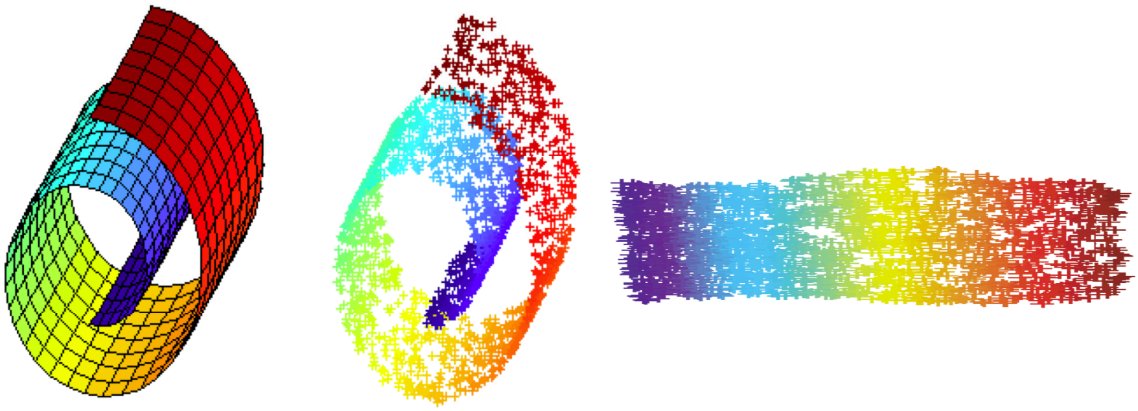

Figure 5.1: A taxonomy of dimensionality reduction techniques. Figure taken from van der Maaten et al. (2008).

representation. Among the family of dimensionality reduction approaches, manifold learning techniques assume that the input data $x_i$ do not reside randomly in $\mathbb{R}^D$, but rather approximately on a manifold denoted by $\mathcal{M}$ [Lin and Zha (2008)].

In this chapter, manifold learning refers to the set of nonlinear dimensionality reduction methods with assumption that input data are sampled from a smooth manifold. The manifold can be simply a hyperplane (linear manifold) or more complicated. An example of a 'curved' manifold known as "Swiss roll" is illustrated in Figure 5.2 combined with the data points lying on it and the projected data onto lower-dimensional space using manifold embedding techniques.

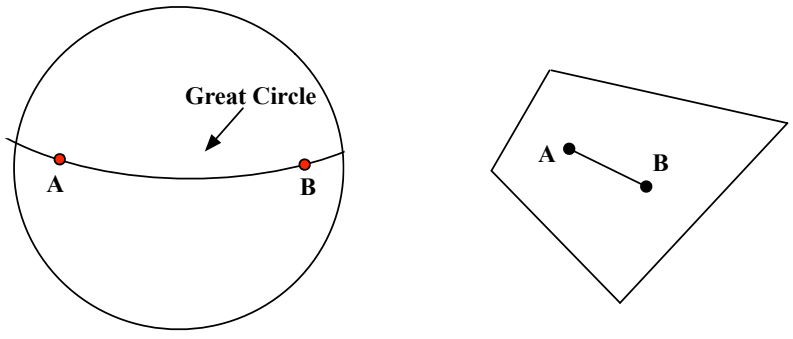
The path to connect two points $x_i$ and $x_j$ in $\mathcal{M}$ can be defined in multiple ways. The shortest path is one way known as the 'geodesic' between $x_i$ and $x_j$ [Law (2006)]. Geodesics on spheres are curves with zero covariant derivatives of their velocity vectors along the curve. For example, the geodesic between two points on a sphere is a part of a "great circle", *i.e.* it is a circle whose centre is synchronised with the centre of the sphere. The straight line in the sphere is actually a curve in the Euclidean perspective (Figure 5.3). The geodesic distance is then defined between $x_i$ and $x_j$ as the length of the geodesic between them.

Most of the nonlinear mapping algorithms that are manifold-based define a neighborhood $Neighbor(x_i)$ of the data point $x_i$. Two types of neighborhood construction



**(1) Swiss roll** **(2) Data points** **(3) Unfolded data**

Figure 5.2: The problem of the manifold illustrated with the example of a curved manifold known as the "Swiss roll". (1) Surface of the manifold on the high-dimensional space. (2) Data points reside on the manifold. (3) The projected data in the lower-dimensional space. Figure taken from [Boyd and Vandenberghe (2004)].




**(a) Geodesic distance** **(b) Euclidean distance**

Figure 5.3: The difference between the shortest path in Euclidean and spherical spaces. The distance between two points $A$ and $B$ can be expressed using the shortest path and known as the geodesic between them. (a) The shortest distance or geodesic between two points is curved in spherical geometry and represents part of a great circle. (b) From the Euclidean perspective the shortest path is a straight line between two points.

are commonly used:

- In $\varepsilon$-neighborhood, the $x_j \in Neighbor(x_i)$ is satisfied if $\|x_i - x_j\| < \varepsilon$, where the $\|.\|$ is the Euclidean distance between $x_j$ and $x_i$ in $\mathbb{R}^D$.
- In kNN, the $x_j \in Neighbor(x_i)$ is satisfied if $x_j$ is one of the $k$ nearest neighbours of $x_i$. Using this construction has the advantage of being independent of the scale of the data; thus it can generalise to small neighborhood size even with a large number of data points.

### 5.2.2 Linear Techniques

Linear techniques achieve the dimensionality reduction by embedding the input data into a subspace of lower-dimensionality. Various techniques have been proposed to do so; however, PCA is by far the most popular. The following sections discusses PCA and Linear Discriminants Analysis (LDA) as examples of the linear techniques.

#### 5.2.2.1 Principal Component Analysis (PCA)

PCA is a statistical technique that analyses a set of data with correlated variables and maps it to a set of uncorrelated variables called principal components (PCs) [Smith (2001)]. Mathematically, PCA defines a linear mapping $\mathcal{M}$ that maximise $\mathcal{M}^T\ cov(X)\mathcal{M}$, where $cov(X)$ is the covariance function of $X$ [van der Maaten et al. (2008)]. This mapping is formed by the $U$ principal eigenvectors (expressed as principal components) of the covariance matrix as following:

$$cov(X)\mathcal{M} = \lambda\mathcal{M}, \tag{5.1}$$

where $\lambda$ represents eigenvalues used to solve the eigenproblem for $U$. The lower-dimensional representation $y_i \in Y$ is defined for the data points $x_i \in X$ by mapping them onto the linear basis $\mathcal{M}$, *i.e.* $Y = (X - \bar{X})\mathcal{M}$, where the $\bar{X}$ is the mean matrix of $X$.

The main drawback of PCA is that the size of the covariance matrix is relative to the dimensions of the data points. Thus, computing the eigenvectors for very high-dimensional data might be infeasible. The computational problem can be overcome for data set with $N < D$ dimensions by using the squared Euclidean distance matrix to compute the eigenvectors instead of the eigenvectors of the covariance matrix($X -$

$\bar{X})(X - \bar{X})^T$. Alternatively, iterative techniques such as probabilistic PCA [Roweis (1998)] may also be used as a solution.

#### 5.2.2.2 Linear discriminant analysis (LDA)

Linear Discriminant Analysis (LDA) [Swets and Weng (1996)] searches for the linear vectors in the underlying space that best distinguish between classes, rather than those that best describe the data. Given a $D$-dimensional data feature, LDA defines a linear combination of these that generates the largest mean differences between the classes [Martinez and Kak (2001)]. Mathematically, for all the samples of all classes, two measures are calculated. The first is the within-class scatter matrix as follows:

$$S_w = \sum_{j=1}^{c} \sum_{i=1}^{F_j} \left(x_i^j - \mu_j\right) \left(x_i^j - \mu_j\right)^T \tag{5.2}$$

where $x_i^j$ is the $i$th feature of class $j$, $\mu_j$ is the mean of class $j$, $c$ is the number of classes, and $F_j$ the number of features in class $j$. The second one is the between-class scatter matrix defined as:

$$S_b = \sum_{j=1}^{c} \left(\mu_j - \mu\right) \left(\mu_j - \mu\right)^T, \tag{5.3}$$

where $\mu$ represents the mean of all classes.

The LDA goal is to minimise the within-class measure, while maximising the between-class measure. This is achieved by maximising the ratio of the determinant of between-class $S_b$ to the determinant of within-class $S_w$ as $\frac{\det|S_b|}{\det|S_w|}$. One problem is that there are at most $c-1$ non-zero eigenvectors; thus, an upper bound of $d$ is $c-1$, while defining $S_w$ needs $D+c$. To solve this, an intermediate space is firstly defined using PCA, then the final $d$-dimensional space is generating using LDA.

### 5.2.3 Nonlinear Techniques

Section 5.2.2 reviewed some of the dimensionality reduction linear techniques, which have been established and well studied in the literature. In contrast, most of the non-linear dimensionality reduction techniques have been presented more recently and are therefore less well studied. In this section, we discuss six nonlinear techniques, subdivided into two main categories. The basic assumption in the dimensionality reduction techniques is that the input data lie on or close to a smooth low-dimensional manifold

| Technique | Key idea | Key computation | Manifold? |
|---|---|---|---|
| KPCA | PCA in feature space | Eigenvectors of the matrix | No |
| MDS | Preserves all inter-point distances, and model similarity and dis-similarity data | Computes pairwise euclidean distance then eigenvectors of the matrix | No |
| Isomap | Preserves geodesic distances | Computes shortest path then eigenvectors of the matrix | Yes |
| LLE | Preserves linearity and reconstructs weights | Computes the weights for each point, then solves the embedding as eigenfunctions. | Yes |
| Laplacian eigenmap | Preserves the locality | Construct a Laplacian graph then solve the embedding as eigenfunctions | Yes |
| Hessian LLE | Locally isometric to a connected group of points | An extension of Laplacian eigenmaps by replacing the Laplacian with the Hessian. | Yes |

Table 5.1: A comparison of nonlinear mapping techniques reviewed in this chapter. The dimensions of the low-dimensional space is assumed to be pre-defined, though some of the techniques can estimate it. If the technique is inspired from the notion of a manifold, the 'Manifold' column has an entry of 'yes' [Lin and Zha (2008)].

[Lin and Zha (2008)]. Each technique attempts to preserve a different property of the underlying structure. Techniques in the first category – global approaches like Kernel PCA, MDS and Isomap – attempt to preserve the metrics at all scales, which gives a more faithful embedding (Section 5.2.3.1). On the other hand, techniques in the second category – local approaches such as LLE, Laplacian eigenmap and Hessian LLE – attempt to preserve the proximity relationships within the data (Section 5.2.3.2). The low-dimensional embedding is reduced to solve a sparse eigenproblem under a covariance constraint. However, this constraint causes a loss of the aspect ratio and consequently these techniques cannot reflect the underlying structure of the global shape of the embedding data. Table 5.1 presents a comparison adapted from [Lin and Zha (2008)] between the nonlinear techniques and reviewed in the following sections.

#### 5.2.3.1 Global Nonlinear Techniques

The global techniques for dimensionality reduction retain the global properties of each data point in dataset [van der Maaten et al. (2008)]. The following sections present KPCA, MDS and Isomap as examples of global nonlinear techniques.

- **Kernel PCA**

KPCA is an extension of traditional linear PCA, constructed in the high-dimensional space using a kernel function [Schölkopf et al. (1998)]. Unlike the PCA that computes

the principal eigenvectors from the covariance matrix, KPCA uses the kernel matrix [van der Maaten et al. (2008)]. It is straightforward in the kernel space, due to the similarity between the kernel matrix and the in-product of the data points in the high-dimensional space constructed using the kernel function. Reformulating PCA in the kernel space helps KPCA to construct nonlinear mappings. KPCA calculates the kernel matrix $K$ of the data points $x_i$, with entries defined as:

$$k_{ij} = \ker\left(x_i, x_j\right), \tag{5.4}$$

where $\ker(\cdot\,,\cdot\,)$ is a kernel function [Spearman (1904)]. Then the principal eigenvectors $v_i$ with eigenvalues $\lambda_i$ of the kernel matrix are computed, and the eigenvectors of the covariance matrix $\alpha_i$ are computed as follows:

$$\alpha_i = \frac{1}{\sqrt{\lambda_i}} X v_i. \tag{5.5}$$

The data point $x_i$ is then projected onto the eigenvectors of the covariance matrix $\alpha_i$, generating the low-dimensional data representation $y_i$:

$$y_i = \left\{ \sum_{j=1}^{N} \alpha_1^j \ker\left(x_j, x_i\right), \ldots, \sum_{j=1}^{N} \alpha_d^j \ker\left(x_j, x_i\right) \right\} \tag{5.6}$$

where $\alpha_1^j$ is the $j$th value in the vector $\alpha_1$. The performance of KPCA is affected by the kernel function used to solve the problem. An important drawback of KPCA is that the size of the kernel matrix is relative to the square of the number of points in the dataset.

- **Multidimensional Scaling (MDS)**

MDS is a group of statistical methods used to visualise the underlying structure of data point relationships using a geometrical representation. Originally, it started from the psychology area and the study of human perception of similarity and dissimilarity between pairs of entities. MDS has since been adopted in various disciplines as a data analysis tool. The concept involves describing each object or entity as a point in the output space. Then a set of input points is arranged in such a way that the distances between pairs of points reflect the degree of similarity between original objects. This means that the MDS represents input data in a new space, where two similar entities are translated as two adjacent points and two different entities are translated as two distant points. This new space is a low-dimensional Euclidean or non-Euclidean space.

Mathematically [Cox and Cox (2000)], calculating the Euclidean distance $d_{ij}$ between a pair of points $i$ and $j$ in a given dimension $a$ can be written for the classical MDS as:

$$d_{ij} = \sqrt{\sum_{a=1}^{D}(x_{ia} - x_{ja})^2}. \tag{5.7}$$

Following the Minkowski model, which is equal to the Euclidean distance when the number of dimensions $p = 1$ or 2, equation( 5.7) can be written as:

$$d_{ij} = \left[\sum_{a=1}^{N} |x_{ia} - x_{ja}|^p\right]^{\frac{1}{p}} \tag{5.8}$$

where $x_{ia}$ and $x_{ja}$ are the coordinate values of the points $i$ and $j$ respectively, $i, j = 1, 2, \ldots, N$ where $N$ represents the number of objects, $a = 1, 2, \ldots, D$ with $D$ represents the dimensions number and $\frac{1}{p} \leq 1$ is defined by the experimenter. The mapping or the representation quality is tested using stress functions such as the row stress function [van der Maaten et al. (2008)]. This computes the error between the pairwise relation of the object in the input space and the pairwise distance in the output space. The row stress function can be expressed as:

$$\boldsymbol{\phi}(Y) = \sum_{ij} \left(\|x_i - x_j\| - \|y_i - y_j\|\right)^2, \tag{5.9}$$

where $\|x_i - x_j\|$ is the Euclidean distance between two points of the high-dimensional data set $X$ and $\|y_i - y_j\|$ is the Euclidean distance between the representative points in the low-dimensional data set $Y$.

- **Isometric Feature Mapping (Isomap)**

Isomap is a graph-based technique that compounds the highlighted features of the PCA and MDS and guarantees asymptotic coverage and computational efficiency [Tenenbaum et al. (2000)]. Compared with other dimensionality reduction techniques, Isomap finds a representative global solution that has the ability to explore the underlying degree of freedom for complicated data such as handwritten characters and biomedical data. It builds on the MDS that depends on the Euclidean distance, which

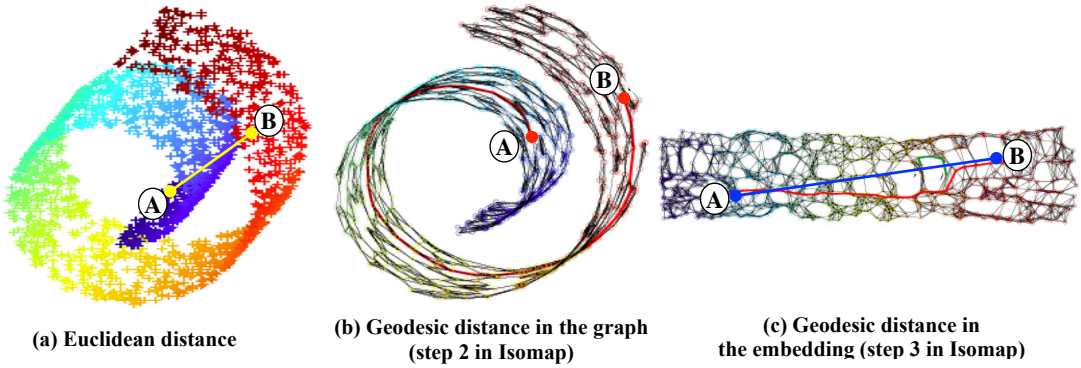



Figure 5.4: The Isomap interpretation for the Swiss roll dataset using geodesic distance. (a) The distance between two points (yellow line) using the Euclidean distance. (b) Constructing the neighborhood graph in the first step of the Isomap (with $k = 7$ for data points with $N = 1,000$) defines an approximation (red segments) of geodesic distances to be computed in the second step. (c) The two-dimensional embedding constructed by the Isomap in the third step, with the true geodesic (blue line) approximated by the shortest path (red line). Figure taken from [Yang (2005)].

ignores the distribution of the datapoint neighbourhood and results in missing the intrinsic dimensions of the data. For example, in the high-dimensional space of Swiss roll dataset, two data points can be considered by MDS as neighbours, although there are longer distances in the manifold than in the actual interpoint distance [Tenenbaum (1998)]. Isomap overcomes this problem by considering the geodesic distance, which is the shortest path or curve that connect two points in a connected set.

The algorithm for Isomap has three phases [Tenenbaum et al. (2000)]. Firstly, for each data point $\boldsymbol{x}_i$, $k$ nearest neighbours $\boldsymbol{x}_j$ are detected using the geodesic distance $d_x(i,j)$. These distances are used to create the weighted graph $G$, with each node $\boldsymbol{x}_i$ connected to its neighbour nodes by the $d_x(i,j)$ edges. Secondly, using Dijkestra's algorithm, the shortest path $d_G(i,j)$ between each pair of data points is computed in the graph $G$ to (over)estimate the geodesic distances between them. This will generate a matrix $D_G = \{d_G(i,j)\}$ of pairwise geodesic distance between each pair of data points in the manifold $M$. The final step will execute the MDS on the distance matrix to embed the high-dimensional data space $X$ in the intrinsic low-dimensional space $Y$.

Figure 5.4 shows the Isomap representation for the Swiss roll dataset. In the first image (left), two points (circled) in the high-dimensional space can be considered neighbours based on the Euclidean distance (yellow line) that does not reflect the

intrinsic distance in the low-dimensional space as defined by the geodesic distance. The second image (centre) presents the neighbourhood graph $G$ constructed by Isomap with $K$=7 nearest neighbours and $N$=100 data points (geodesic distance is drawn in red line). The final image (right) is the two-dimensional embedding by Isomap, where the true geodesic in the embedding (blue) is approximated by the shortest path (red).

#### 5.2.3.2 Local Nonlinear Techniques

The local techniques for dimensionality reduction embed the data in low-dimensional space by maintaining the local features of each data point [van der Maaten et al. (2008)]. The following sections present LLE, Laplacian eigenmap and Hessian LLE as examples of the local nonlinear techniques.

- **Locally Linear Embedding (LLE)**

LLE is a local technique for dimensionality reduction involving the construction of a graph representation similar to Isomap. In contrast to Isomap, it aims at preserving the local properties of the data points [van der Maaten et al. (2008)]. Following MDS and PCA, LLE is considered an eigenvector method. It captures the local geometry of each point by computing the coefficients of its neighbour points. These coefficients define the local features of the manifold near each point $\boldsymbol{x}_i$. The first step is to find the nearest neighbourhood points for each data point $\boldsymbol{x}_i$, and then define the linear combination weight factor $w_{ij}$ for the point $x_i$ with its neighbour points $\boldsymbol{x}_j$. As a consequence, finding the low-dimensional embedding $\boldsymbol{y}_i$ for each $\boldsymbol{x}_i$ using the $w_{ij}$ amounts to minimising the cost function:

$$\phi\left(y\right) = \sum_{i} \left( \boldsymbol{y}_i - \sum_{ij} w_{ij} \boldsymbol{y}_j \right)^2 . \tag{5.10}$$

- **Laplacian eigenmap**

A Laplacian eigenmap [Belkin and Niyogi (2003)] constructs an orthogonal smooth function defined on the manifold using the Laplacian of the neighborhood graph. Unlike Isomap that represents the distance between $x_i$ and $x_j$ as the weight $w_{ij}$ of the edge $(x_i, x_j)$, a Laplacian eigenmap uses the similarity between $x_i$ and $x_j$ as the weight, defined as follows [Law (2006)]:

The process starts by constructing a neighbourhood graph of $Y$ by either the $\epsilon$-neighborhood or the kNN neighbourhood. Then the weight $w_{ij}$ is defined as follows:

$$w_{ij} = exp\left(-\frac{\|x_i - x_j\|^2}{4t}\right), \tag{5.11}$$

where $t$ is an algorithmic parameter, or it can simply equal one. The Laplacian graph $L$ is computed by $L = D - W$, where $W = w_{ij}$ is graph weight matrix computed in equation 5.11, and $D$ with $d_{ij} = \sum_j w_{ij}$ is the diagonal matrix. Finally, the generalised eigenvalue $Lu = \lambda Du$ is solved and the second to the $(d+1)$-th smallest eigenvalues is defined. The eigenvectors $u_1, \ldots, u_d$ are then used as the low-dimensional feature vectors.

- **Hessian LLE (HLLE)**

Hessian LLE (HLLE) [Donoho and Grimes (2003)] is a reformulation of LLE to minimise the 'curviness' of the high-dimensional space when transferred to a low-dimensional space, with the constraint that the low-dimensional representation is locally isometric [van der Maaten et al. (2008)]. It starts by applying the Euclidean distance to identify the kNN neighbours for each data point $x_i$ . In the neighborhood, local linearity of the manifold is assumed. PCA is then applied on its $k$ nearest neighbours $x_{ij}$ by finding the $d$ principal eigenvectors $M = m_1, m_2, \ldots, m_d$ of the covariance matrix $cov(x_i)$. A matrix $Z_i$ is then constructed, containing all cross products of $M$ up to the $d$th order, and is orthonormalised by applying the Gram-Schmidt orthonormalisation procedure [Arfken and Weber (2005)]. An estimator for the Hessian $H^l$ at the point $x_i$ is defined as the transpose of the last $\frac{d(d+1)}{2}$ columns of the matrix $Z_i$. A matrix $\mathcal{H}$ is finally constructed with entries:

$$\mathcal{H}_{i,j} = \sum_l \sum_r \left(\left(H^l\right)_{r,i} \times \left(H^l\right)_{r,j}\right), \tag{5.12}$$

where $H^l$ is, again, the $\frac{d(d+1)}{2}$ by $k$ matrix defined as the estimator for the Hessian over the neighbourhood $l$, and the row $r$ is a specific entry in the Hessian matrix and column $i$ is a specific point at the neighbourhood matrix. The matrix $\mathcal{H}$ contains the information of the high-dimensional data manifold curviness . The low-dimensional data representation is defined as the eigenanalysis of $\mathcal{H}$, which correspond to the d smallest non-zero eigenvalues of $\mathcal{H}$.

## 5.3 Spatio-temporal Graph-based Isomap

Section 5.2 reviews the existing methods in the dimensionality reduction and manifold embedding domain. These methods have been applied to the video-processing tasks, nevertheless none of them consider the spatio-temporal correlation in the content. The aims were visualisation, reduction of the features dimension or data compression. To consider the temporal information, most of the works applied pre-processing techniques such as template matching, camera calibration analysis or object tracking, which required more time to compute and space to store.

To overcome these shortcoming, this stage delivers a spatio-temporal graph-based manifold embedding technique in two phases. First, content of the video stream is described in the high-dimensional space using the invariant interest points and coding schemes, which have been presented in detail in Chapter 3 and Chapter 4 and were briefly restated in Section 5.3.2. The video volume is firstly reduced to a set of interest points extracted from the spatio-temporal domain using ST-SIFT, then the spatio-temporal codes are generated using a ST-LLC which considers the locality of the manifold structure in the input space. During this phase the intra-correlation is computed between the extracted features and the codes using the spatio-temporal kNN graph. Second, a manifold is computed to map the high-dimensional representation to the embedded space (Section 5.3.3). During this phase the inter-correlation is computed between the multiple retakes using the spatio-temporal kNN graph. The spatial Isomap [Tenenbaum et al. (2000)] was extended to the spatio-temporal domain to generate the set of coordinates for each manifold. Generated coordinates were chronologically ordered by the spatio-temporal similarity and integrated to a graph for sequences alignment. The entire process of the approach is illustrated in Figure 5.5.

### 5.3.1 Notation

The following notation is used in the following section of this chapter. Let $X = \{x_1, \ldots, x_N\} \in \mathbb{R}^{D \times N}$ denote a video sequence containing repetitive contents with $N$ frames and $D$ dimensions, and $x_i$ represent a frame in $X$. $F = \{fx_1, \ldots, fx_N\} \in \mathbb{R}^{Q \times N}$ is the ST-SIFT matrix with $N$ columns and $Q$ dimensions, where $fx_i = \{f_1, \ldots, f_M\}$ represents a vector of $M$ features for frame $x_i$. Let $S = \{sx_1, \ldots, sx_N\} \in \mathbb{R}^{U \times N}$ be the spatio-temporal codes matrix that represent the video frames with $N$ codes and $U$ dimensions, and $sx_i = \{s_1, \ldots, s_H\}$ represent a set of $H$ codes for the frame $x_i$.

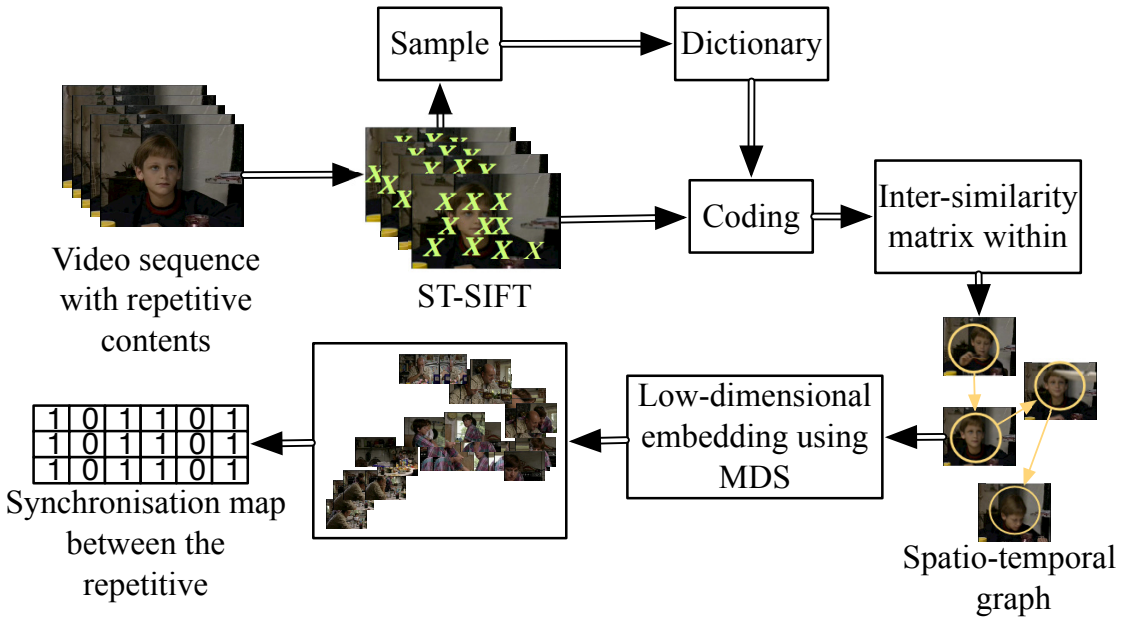



Figure 5.5: Processing steps for spatio-temporal alignment of nearly-repetitive contents in a video stream. Firstly, the video representation is defined by extracting the spatio-temporal invariant interest points using ST-SIFT followed by generating a local linear codes using ST-LLC. Secondly, the similarity is measured by constructing a shortest-path graph with kNN in both the spatial and the temporal domains. Finally, the graph-based representation is mapped to a lower-dimensional space using MDS.

### 5.3.2 Video Representation

The first step is to apply the ST-SIFT algorithm that identifies spatially and temporally invariant interest points in a given video stream. These points contain the amount of information sufficient to represent the video contents. To achieve the invariance in both space and time, spatio-temporal Gaussian and DoG pyramids are calculated. The common points $F$ between the three spatial and temporal planes ($xy$, $xt$ and $yt$) at each scale in the DoG are chosen as interest points.

The second step is to derive the spatio-temporal codes $S$ for the video stream $X$ given the ST-SIFT feature matrix $F$. For each frame $x_i$, the algorithm works by firstly constructing a spatio-temporal graph between its descriptor set $fx_i$ and a codebook $B$, computing the shortest path, performing a kNN search, and finally solving the

following constrained least-square fitting problem:

$$\min_{sx_i} \sum_{j=1}^{N} \|fx_j - Bsx_j\|^2 + \lambda \|d_j \odot sx_j\|^2 \quad st. \quad 1^\top sx_j = 1, \ \forall j$$

where $\odot$ is the element-wise multiplication, $\lambda$ is a sparsity regularisation term and $d_j$ is the locality parameter that represents every basis vector with different freedom based on its shortest path to the spatio-temporal descriptor. '$1^\top sx_j = 1, \ \forall j$' is the shift-invariant requirement for the LLC code.

The final step uses multi-scale max pooling [Serre et al. (2005)], where the set of codes computed for each frame are grouped together to create the corresponding pooled representation $S$.

### 5.3.3 Manifold Embedding

Given a spatio-temporal coding matrix $S$ for a video sequence $X$, the synchronisation map is estimated between the multiple video retakes. First the spatio-temporal kNN graph $\delta$ is constructed using the Euclidean distance between the LLC codes. Each node represents a spatio-temporal code for one frame, while each edge represents a connection between two frames. The value of $\delta_{ij}$ defines the distance between the LLC codes $sx_i$ and $sx_j$ for two frames $x_i$ and $x_j$ $(i, j = 1, \ldots, N)$. Then, as illustrateded in Figure 5.6, for each frame instance $x_i$ represented by a code $sx_i$:

1. $k$ frames whose distance is the closest to $x_i$ are connected. They are referred to as spatial neighbours $(sn)$ (Figure 5.6(1)):

$$sn_{x_i} = \left\{ sx_{j1}, \ldots, sx_{jk} \mid \underset{j}{\text{argmin}}^k(\delta_{ij}) \right\} \tag{5.13}$$

   where $\underset{j}{\text{argmin}}^k$ implies node indexes $j$ with smallest distances.

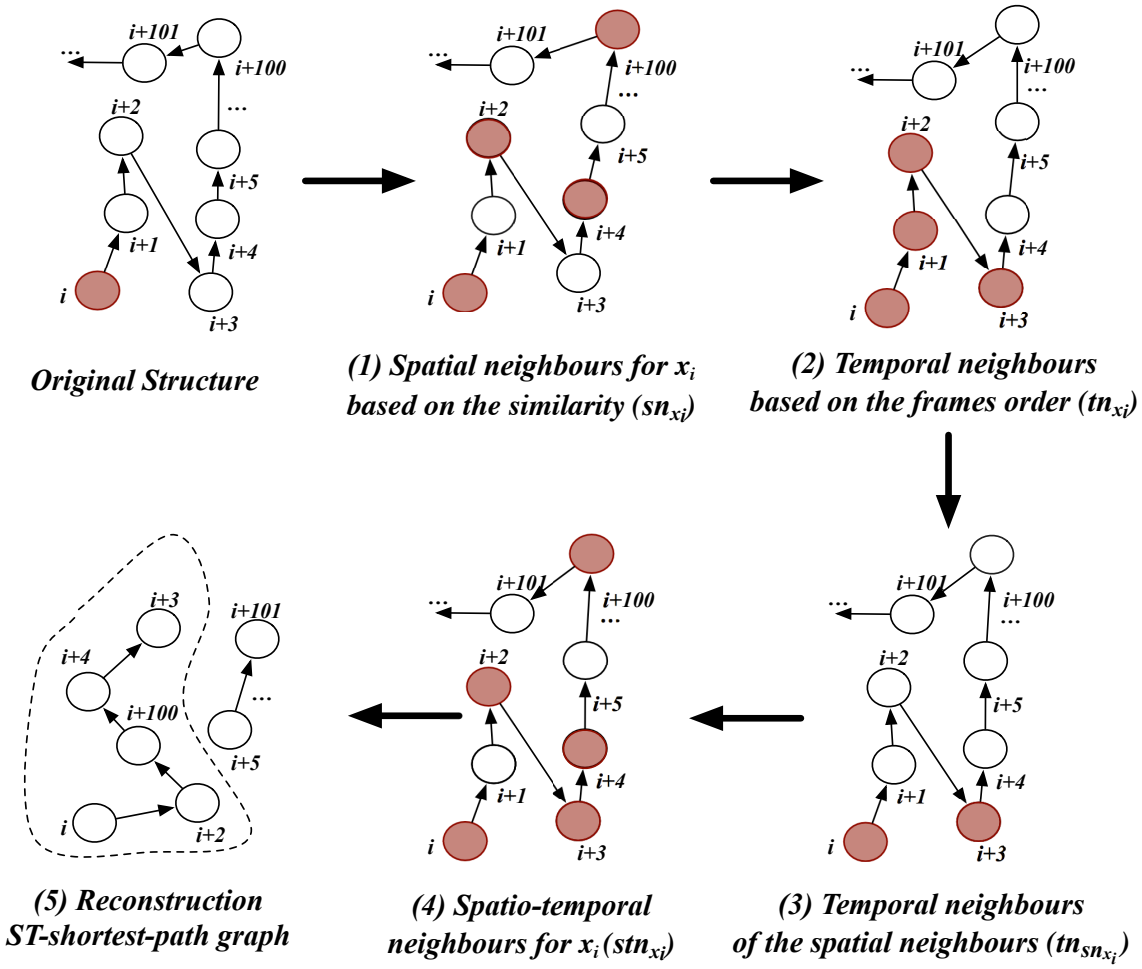



Figure 5.6: The construction of the spatio-temporal distance graph. The initial graph is firstly derived for $x_i$ using the kNN method. (1) spatial neighbours are then defined based on the similarity, (2) temporal neighbours are defined based on the frames order within the video, (3) temporal neighbours are defined from temporal neighbours of spatial ones, (4) union between the (1) and (3) creates the spatio-temporal neighbours. Finally (5) the geodesic distance is recalculated by finding the shortest-path using the Dijkstra algorithm.

2. Another $k$ frames, chronologically ordered around $x_i$, are set as temporal neighbours ($tn$) (Figure 5.6(2)):

$$tn_{x_i} = \left\{ sx_{i-\frac{k}{2}}, \ldots, sx_{i-1}, sx_{i+1}, \ldots, sx_{i+\frac{k}{2}} \right\} \tag{5.14}$$

3. To optimise the set of temporal neighbours, $tn_{sn}$ is selected from temporal neigh-

bours of spatial neighbours (Figure 5.6(3)):

$$tn_{sn_{x_i}} = \left\{ tn_{x_{j1}} \cup \ldots \cup tn_{x_{jk}} \right\} \cap tn_{x_i} \tag{5.15}$$

4. Spatial and temporal neighbours are integrated, producing spatio-temporal neighbours ($stn$) for frame $x_i$ (Figure 5.6(4)):

$$stn_{x_i} = sn_{x_i} \cup tn_{sn_{x_i}} \tag{5.16}$$

   The above formulation of $stn_{x_i}$ effectively selects $x_i$'s temporal neighbours that are similar, with a good chance, to its spatial neighbours. This means that, supposing $x_i$ is an isolated frame and totally different from the temporal neighbours, only the spatial neighbours will be taken into consideration.

The inter-correlation matrix is constructed by recalculating the shortest path between the nodes in graph $\delta$, forming a new embedded correlation $\delta_\gamma$ as shown in Figure 5.6(5).

The manifold embedding is then modelled as a transformation $T$ of the high-dimensional data in terms of similarity $\delta_\gamma$ into a new embedded configuration $E$ in the low-dimensional space [Borg and Groenen (2007)]:

$$T : \delta_\gamma \ \rightarrow E \tag{5.17}$$

Using the classical MDS method defined by Borg and Groenen (2007), the function $T$ is solved by minimising the *Strain* loss function [Carroll and Chang (1970)] that is used to find the coordinates from a matrix of squared distances such as Euclidean distance or shortest path distance and is defined as:

$$\begin{aligned} L_{embedding} &= \|X - T(X)\| \\ &= \|X - T(\delta_\gamma)\| \\ &= \|X - (Q \wedge Q^T)\| \\ &= \|X - (Q_+ \wedge_+^{\frac{1}{2}})\| \end{aligned} \tag{5.18}$$

$\delta_\gamma$ is decomposed using the eigenvectors $Q$ and the eigenvalues $\wedge$, where $\wedge_+^{\frac{1}{2}}$ contains the $e$ largest eigenvalues in $\wedge$ along the diagonal, and $Q_+$ is the square root of $e$ columns in $Q$. The new coordinates for each frame instance in the embedded space are selected from the $e$ largest eigenvalues of matrix $Q_+ \wedge_+^{\frac{1}{2}}$.

## 5.4 Experiments

The approach was evaluated using MPEG-1 videos from the NIST TRECVID 2008 BBC rushes video collection [Over et al. (2008)]. The aim of this experiment was the identification of repeated contents in a video stream. Therefore, the nature of the dataset is an important factor. The rushes video collection was the most suitable (*c.f.* Chapter 2, Section 2.4), as it contains repetitive contents from multiple retakes of the same scene, caused by, for example, actors' mistakes or technical failures. Nearly-repetitive contents in the rushes video may not be identical, sometimes causing inconsistency between retakes. Occasionally some parts of the original sequence are dropped or extra information may be added at various places, resulting in retakes of the same scene with unequal lengths. Figure 5.7 shows sample scenes with their retakes from one of the rushes videos. Five video sequences were selected containing drama productions in the following genres: detective, emergency, police, ancient Greece and historical London. In total an approximate duration of 82 minutes was gathered with frame size of $288 \times 352$ pixels.

For video representation, the same settings used in Chapter 4 are adopted in this experiment. Frames segmentation is firstly applied to generate a sequence of frames. Spatio-temporal interest points are detected from the segmented frames using ST-SIFT as developed in Chapter 3. The spatio-temporal regions around the interest points are described using the 3D-HOG descriptors [Scovanner et al. (2007)]. The ST-LLC codes presented in Chapter 4 were then computed for each spatio-temporal sub-region and pooled together using multi-scale max pooling [Serre et al. (2005)] to create the corresponding pooled representation. The experiment used $4 \times 4$, $2 \times 2$ and $1 \times 1$ sub-regions. The pooled features were then concatenated and normalised using the $\ell^2$-norm. The initial number of frames to be spatially and temporally connected in the manifold appeared dependent on the clip length and was selected manually.

### 5.4.1 Ground truth construction

Each scene from the rushes videos consists of a few actions, sometimes including actors' dialogue. They were used as units of evaluation and the purpose of the experiment was to group and align the multiple similar retakes of the same scene. Figure 5.8 shows examples for scene descriptions. The description of actions for each scene was provided by the NIST for the BBC rushes video summarisation task in 2008 Over et al. (2008). The ground truth was constructed for each video in a fashion similar the procedure

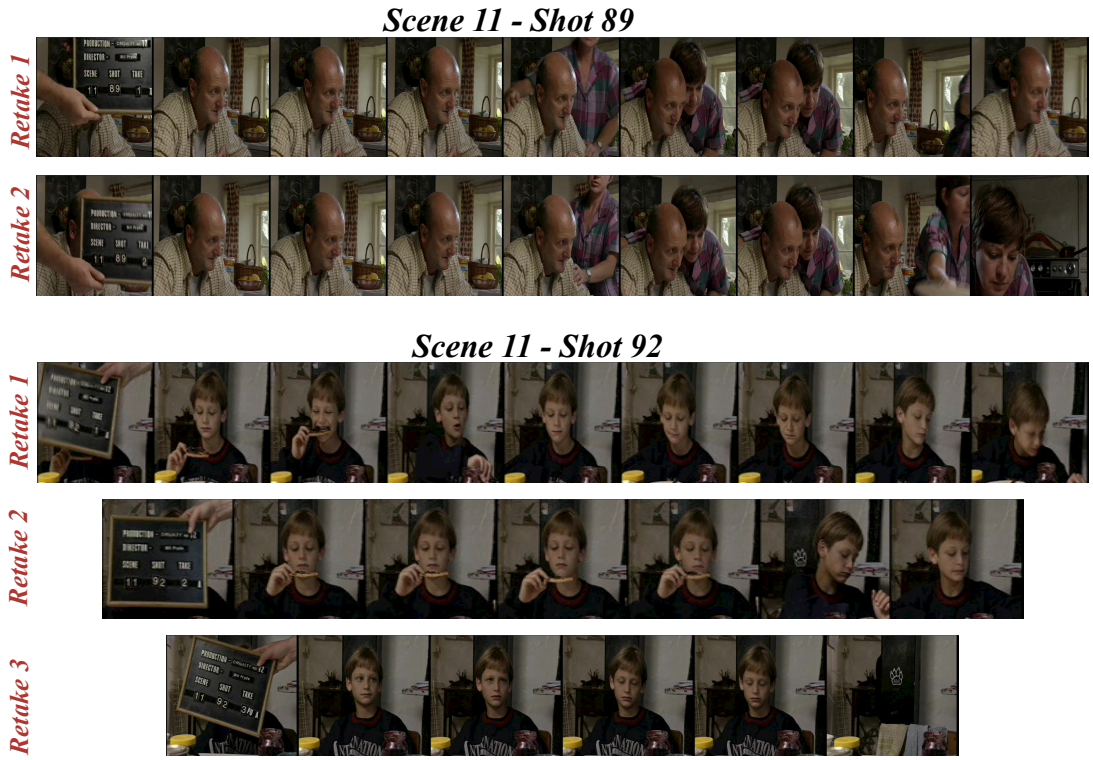



Figure 5.7: Frame samples from the rushes video identified as *MRS044499* in TRECVID 2008. Two different shots were recorded at the same indoor scene. The first shot 89 from scene 11 involves a man and a woman in a kitchen, recorded twice with some variations. The second shot 92 from scene 11 consists of a child eating in a kitchen, recorded three times with some variations and different length as well.

presented by Chantamunee and Gotoh (2010), where three human judges are used to define repetitive content at a frame rate of 0.5 fps (one frame per two seconds). The judges were asked to study the video summary and use it to identify the start and the end for each retake. This resulted in a list of repeated scenes identified by the frame numbers The defined positions for five videos, totalling 39 scenes and 94 retakes, were used as the ground truth. Table 5.2 provides further details of the dataset.

### 5.4.2 Evaluation Schema

The evaluation was based on two types of comparisons: first, comparison with three other frameworks; second, comparison with ST-Isomap [Jenkins and Matarić (2004)] as Isomap extension and other conventional approaches.

In the first evaluation, the approach was compared with the three other simplified

.
..
Scene 3:
Woman lifts child out of chair and walks away to the right.
Scene 4:
Woman facing camera with child in her arms, talking; table with food in front of her.
Woman with child exists to right.
Scene 5:
Child sitting on table and eating, facing the camera.

Figure 5.8: Scene descriptions from a rushes video frame sequence, identified as *MRS044499*. The original description provided by NIST was split into five scenes, of which the last two were shown. The 5th scene starts when a woman stops walking and facing camera.

alternatives (Frameworks 1, 2 and 3). The purpose of the evaluation was to firstly measure the effect of the considering the spatio-temporal information in the framework, then followed by measuring the effect of each stage in the performance.

**Framework 1:** SIFT/LLC/ Isomap.

This approach evaluates the performance of the developed framework that consider the spatio-temporal information. In this case the performance of the spatial techniques that that have been extended during the development of the framework should be considered. It combined the original 2D SIFT by Lowe (2004), LLC coding with the Euclidean distance graph by Wang et al. (2009) and spatial Isomap by Tenenbaum et al. (2000).

**Framework 2:** SIFT/ LLC/ STG-Isomap.

ST-SIFT of the presented work was replaced with the conventional 2D SIFT and ST-LLC coding with spatial LLC based on the Euclidean distance graph. This approach evaluates the impact of the video representation covered by ST-SIFT and LLC with the shortest path graph. This step is to measure the effect of the considering the temporal information on the manifold embedding stage only.

**Framework 3:** ST-SIFT/ ST-LLC/ Isomap.

This approach evaluates the performance of the inter-correlation step covered by

| video id | duration (min:sec) | #scenes | #retakes (#retakes/scene) |
|---|---|---|---|
| *MS206290* | 21:03 | 11 | 27 (2,1,1,3,5,5,2,2,1,4,1) |
| *MS206370* | 12:30 | 7 | 17 (2,2,2,2,3,4,2) |
| *MS215830* | 14:55 | 5 | 14 (3,3,3,2,3) |
| *MRS044499* | 12:42 | 6 | 10 (2,2,2,1,1,2) |
| *MRS1500072* | 21:40 | 10 | 26 (3,5,1,2,2,3,2,3,3,2) |

Table 5.2: Dataset consists of five rushes videos, containing a number of scenes, each with multiple retakes. They are from BBC productions for detective, emergency, police, ancient Greece and London historical dramas

the STG-Isomap and to measure the effect of the considering the temporal information on the video representation stage only. For that ST-SIFT was combined with LLC coding with the shortest path graph and the spatial Isomap.

**Framework 4:** ST-SIFT/ ST-LLC/ STG-Isomap.
The last framework is the developed framework that consider the spatio-temporal information in both video representation and manifold embedding stages. A combination of ST-SIFT and LLC coding with the shortest-path graph was used for intra-correlation, followed by inter-correlation using STG-Isomap.

To measure the performance of frame similarity, precision and recall were calculated. They are the most suitable evaluation criteria, since the numbers of the relevant and the non-relevant videos for each query are pre-defined [Vaughan (2004)]. The aim is to maximise the number of correct predictions while minimising the number of false ones. The assumption was that frames from the same scene would be more similar than those from other scenes. Additionally, frames from different retakes could be nearly identical, while similar frames might not appear sequentially. For example, one actor might walk forward and backward in a scene, thus similar frames would appear in distant positions in the video sequence; also nearly identical frames could appear in different retakes of the same scene. For the video sequence $X$ with $N$ frames the following values are defined:

- $TP_i$: number of detected frames that belong to the same scene with frame $x_i$
- $FP_i$: number of detected frames that do not belong to the same scene with frame $x_i$

- $TN_i$: number of undetected frames that do not belong to the same scene with frame $x_i$
- $FN_i$: number of undetected frames that belong to the same scene with frame $x_i$

The average precision and recall are then calculated:

$$avrage\,precision = \frac{1}{N}\sum_{i=1}^{N}\frac{TP_i}{TP_i + TF_i} \tag{5.19}$$

$$avrage\,recall = \frac{1}{N}\sum_{i=1}^{N}\frac{TP_i}{TP_i + TN_i}. \tag{5.20}$$

The $F$-score is then calculated as final measurement, which is mostly used as a measure of an algorithm's accuracy based on precision and recall:

$$F = 2 \cdot \frac{avrage\,precision \cdot avrage\,recall}{avrage\,Precision + avrage\,recall} \tag{5.21}$$

This can be considered as a weighted average of the precision and recall, with a best score of 1 and worst score of 0. The average precision and recall values of the different frameworks were compared, given the neighbourhood size of $k$ used to construct the spatio-temporal graph. In other words, $k$ was the operating parameter that represented the $k$ closest frames to $x_i$, ranging from 7% to 50% (half) of the number of entire frames, with the step size of 7%.

### 5.4.3 Results

This experiment aimed at the reconstruction of video sequences to uncover their repetitive contents. It applied manifold learning to two types of spatio-temporal correlation matrices. It was hoped that retakes from the same scene would be mapped closer to each other in the manifold, resulting in clusters of repetitive contents. The approach was evaluated using the average precision and recall as defined in Equations ( 5.19) and ( 5.20) followed by computing the $F$-score as presented in Equation( 5.21). The operating parameter $k$ was defined as the size of neighbours, *i.e.* $k$ most similar frames to $x_i$ in both space and time domains.

The question to be answered at this experiment was whether any particular point (frame) in the embedded space had frames from the same scene within the neighbourhood of the size $k$. The value for $k$, the initial number of frames to be connected in the manifold learning, appeared dependent on each video sequence. It might have been

| Video id | best $k$ | Average precision | Average recall | F-score |
|---|---|---|---|---|
| *MS206290* | 28% | 0.71 | 0.77 | 0.74 |
| *MS215830* | 35% | 0.77 | 0.81 | 0.79 |
| *MRS044499* | 35% | 0.82 | 0.80 | 0.81 |
| *MS206370* | 28% | 0.85 | 0.82 | 0.83 |
| *MRS1500072* | 35% | 0.88 | 0.90 | 0.89 |

Table 5.3: The best $k$ for building a neighbourhood graph varied among video sequences. The best precision, recall and $F$-score are shown at the best neighbourhood size $k$.

affected by the length of their repetitive scenes, and was selected manually so that they resulted in the highest precision and recall (Table 5.3).

The best $F$-score for Framework 4 was obtained for video *MRS1500072*. This video contains outdoor scenes with large variations in scene setting such as the appearance of dominant colour patterns, characterised by busy backgrounds and lots of movements by actors and objects. On the other hand, *MS206290* had the worst result. This may have been caused by the nature of the indoor scenes in this video, with crowded people and little movement, so that there were few significant changes between the frames to be captured by ST-SIFT.

Figure 5.9 presents the average precision and recall for each video using the proposed approach and three alternatives. The figure indicates that Framework 4 outperformed other frameworks with a fair margin for all five videos in detecting similar and repetitive scenes. The similarity between the features were firstly captured using intra-correlation, which helped to distinguish between similar scenes recorded, for example, in the same environment. Sharing spatio-temporal features from the common action, actors or environment would cause them to be defined as retakes of the same scene. Intra-correlation deals with this problem by considering the relationship between the features within the frames using ST-LLC. After that, the STG-Isomap measures the relationship between video frames by defining the inter-correlation that reconstructs and reorders the frames order in the lower-dimensional space based on their similarity. Therefore, frames from the same scene were mapped close to each other in the lower-dimensional space, resulting in similar representations in the manifold.

Framework 3 (ST-SIFT/ ST-LLC/ Isomap) performed better than Frameworks 1 and 2 for all five videos. This may be caused by the effect of the intra-correlation within the ST-LLC step. Defining the relationship between the features within all frames allows the framework to locate most of the retakes with high similarity. Using

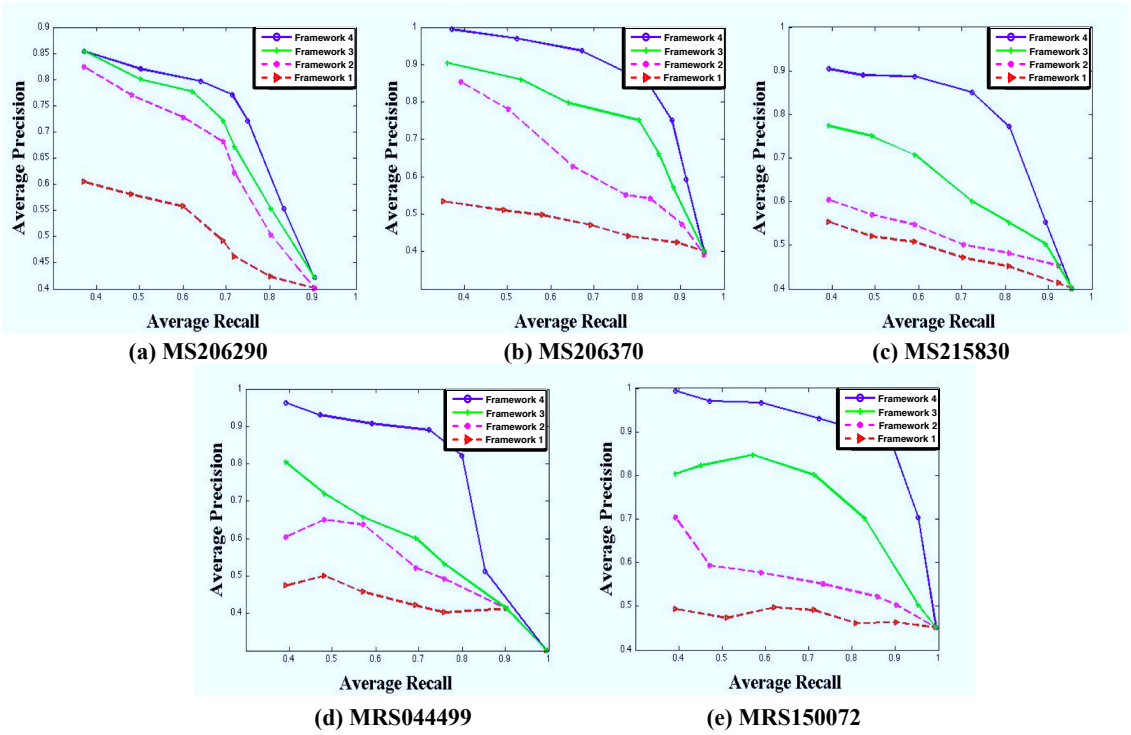



Figure 5.9: Average precision and recall for five rushes videos, identified as *MS206290*, *MS206370*, *MS215830*, *MRS044499*, and *MRS150072*. For each video stream, the spatio-temporal alignment method (blue) is compared with three other alternatives.

the intra-correlation allows it to capture the retakes from the same scene with high similarity. However, not using the inter-correlation causes it to fail in detecting retakes with major changes in the scenes, as shown in Figure 5.9(d), for video *MRS044499*.

Framework 2 (SIFT/ LLC/ STG-Isomap) represents the video contents spatially using feature detector and coding schemes. This combination would only work for the retakes with high similarity in the environment and the actors, which is the case shown in Figure 5.9(a) for video *MS206290*, but not for the actions performed as in Figure 5.9(c) for video *MS215830*. The failure was caused by the variations in scene setting in video *MRS150072*, such as the appearance of dominant colour patterns and the changes of video shooting location within the scene. Such variations had to be captured spatially and tracked temporally.

Framework 1 (SIFT/LLC/ Isomap) exhibited the lowest performance. This was expected since it only considers the spatial information in both content representation and manifold embedding. Some of the scenes, as in Figure 5.9(d) for video *MRS044499*, contain little movement with minor changes in the spatial information; despite the fact

that various camera angles were used, there were not many dramatic changes in the scene (in the foreground or in the background). Ignoring the temporal information in these types of data would define most of the takes from different scenes as retakes for the same scene. In such cases the spatial information is not enough; features had to be captured spatially and tracked temporally and then searched over the entire sequence to detect similar retakes. To sum up the obtained results:

- Intra-correlation was derived by Framework 4 between the codes representation to represent the similarity across the frame sequence in terms of spatial and temporal coherence. This helped to distinguish scenes with different scenarios or events that occurred in the same location, with the same actors, or which shared the same concept (e.g. activities). Similarity measurement between these scenes would cluster them together, which is what happened in the cases of Framework 1 and Framework 2. In contrast, Framework 3 was more able to distinguish such retakes

- On the other hand, inter-correlation is defined in Framework 4 between the sequence of frames and then mapped to the lower-dimensional space. The generated coordinates in the manifold are reordered and clustered, creating a group of repetitive contents based on their spatio-temporal similarity. Some scenes contained a stationary camera or objects movement, captured at multiple locations. Additionally, duration of retakes are different, which causes some of them to contain more objects, locations, or activities than the others. Therefore, they may not share sufficient spatio-temporal features to define their belonging to the same scene. Consideration of the inter-correlation alleviated the above problems, which were handled by Framework 2 better than Framework 3. Framework 2 was more able to distinguish between multiple retakes spatially and then define their spatio-temporal similarity before project them on to the lower-dimensional space.

Figure 5.10 illustrates the reconstruction of video sequences, aiming to uncover their nearly-repetitive contents. Retakes from the same scene were mapped close to each other in the manifold resulting in clusters of repetitive contents. The video sequence *MRS044499* presented in the figure contained six scenes with ten retakes (described earlier in Table 5.2). The left panel of the figure shows the aligned sequences in the 2D space with multiple clusters of frames. Most frames from the same scene were re-positioned and placed close together in the lower-dimensional space. There

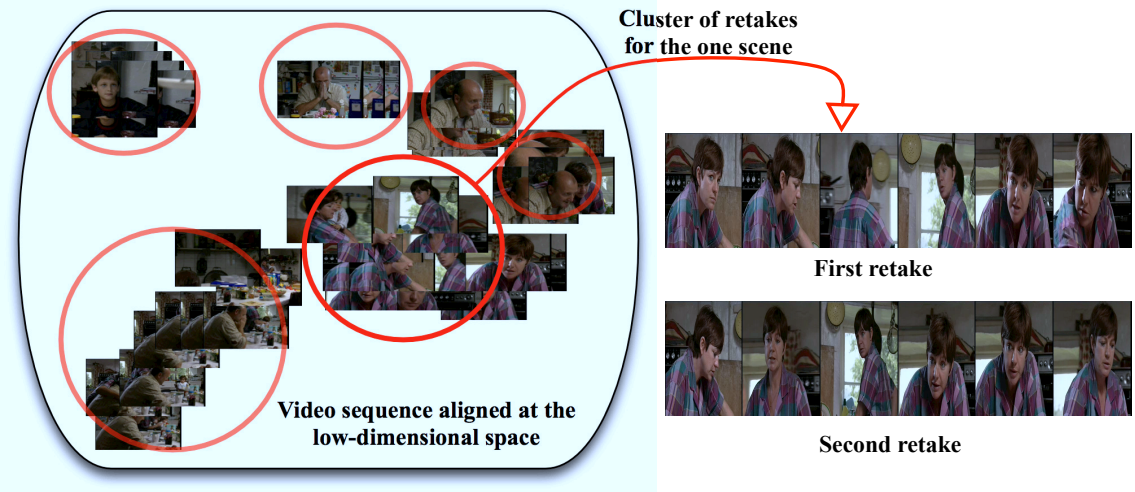
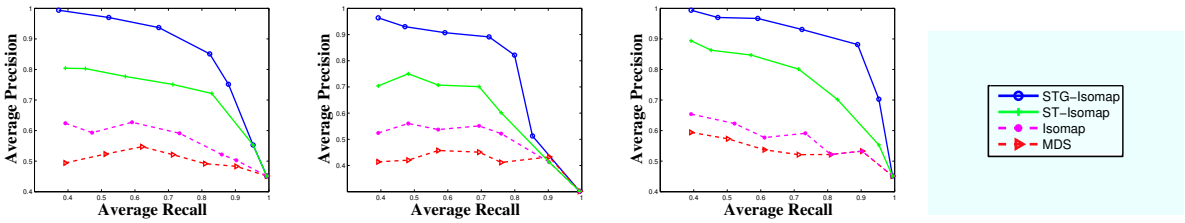



Figure 5.10: Video sequence *MRS044499* was aligned in the two-dimensional space using the neighbourhood size of $k = 15$.



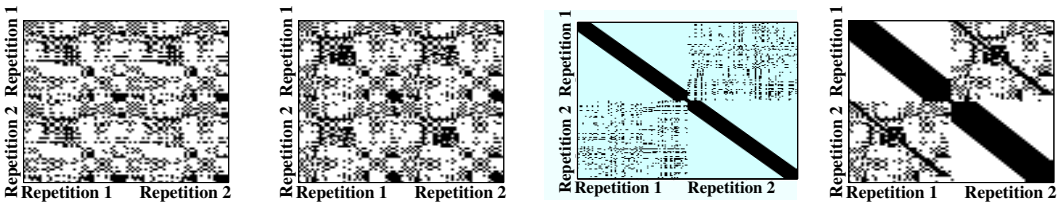

(a)*MS206370* (b)*MRS044499* (c)*MRS150072*

Figure 5.11: Average precision and recall for three rushes videos. The STG-Isomap approach is compared with the MDS, the conventional Isomap and the ST-Isomap.

were many causes, such as camera moves, that could result in discontinuity because such frames did not share sufficient spatial features with others. Consideration of temporal relation in the intra-correlation step alleviated this problem, thus successfully producing a clear video trajectory in the manifold. The contents of one cluster, two retakes of the same scene, are presented in the right panel of the figure.

### 5.4.4 Comparison of STG-Isomap with the State-of-the-Art

Using three videos from the dataset, the average precision and recall were compared with three other approaches including MDS, the conventional Isomap and the ST-Isomap [Jenkins and Matarić (2004)]. Individual performances were evaluated using a kNN graph with the varying value of $k$.




(a)MDS (b)Isomap (c)ST-Isomap (d)STG-Isomap

Figure 5.12: Synchronisation binary map to illustrate the relation between two repetitive sequences in *MRS044499*.

For detection of similar and repetitive scenes, precision and recall scores are presented in Figure 5.11. The figure indicates that the STG-Isomap approach outperformed other methods with all three videos. The inter-correlation helped to capture the similarity of composition between two frames using a spatio-temporal neighbourhood graph. Considering temporal and spatial information separately in ST-Isomap was not sufficient to distinguish between the repetitive contents. Additionally, using temporal windowing serves to capture the temporal history for data points from nearby frames only; this is not the case for repetitive contents that can be distributed non sequentially along the video sequences. Such data with large variations, different camera angles, dramatic changes, colour combination of the foreground and background, *etc.*, require the definition a spatial plus temporal correlation between the frames. On the other hand, the difference was not very large in embedding repetitive manifolds between MDS and Isomap, indicating that these methods could not capture spatio-temporal features very well.

Figure 5.12 illustrates the synchronisation map computed between two nearly-repetitive sequences in rushes video *MRS044499*. The figure shows that the STG-Isomap was able to build cleaner clusters of repetitive contents, with which most frames from the same scene were re-positioned and closely placed in the embedded space. It developed the spatio-temporal relation during neighbourhood graph construction. As a comparison, only the temporal relation was identified by the ST-Isomap while the spatial relation was identified by the Isomap and the MDS. As an additional note, STG-Isomap chose the value for $k$ before the graph was constructed. For Isomap, on the other hand, $k$ was selected during the graph construction, and they could change the value when calculating the shortest paths. The linear embedding technique, MDS, did not have a step for graph construction, but the fixed neighbourhood size were applied. It was not able to learn the structure of repetitive sequences very well.

## 5.5 Conclusion

The previous techniques for video alignment and repetitive scenes detection require frame-to-frame matching with camera calibration and object tracking. This chapter introduced a spatio-temporal graph-based manifold embedding as the third stage of video sequences alignment framework developed throughout this thesis. The similarities observed in frame sequences are captured by defining two types of correlation graphs: an intra-correlation graph between the codes and an inter-correlation graph between the frames.

First, a high-dimensional representation is defined using the space-time invariant interest points detection and coding approaches. The locality constrained spatio-temporal coding technique is applied, which considers the locality of the manifold structure in the input space to define the intra-correlation between the codes. Second, a manifold representation is computed for the video sequence to map the high-dimensional representation to an embedded space. At this stage the inter-correlation is computed between the frames using the spatio-temporal kNN graph.

The approach was evaluated on nearly-repetitive content detection using the rushes video collection from the BBC dataset. The results indicated that the spatio-temporal extension of Isomap performed better than conventional manifold embedding schemes. The contribution of this work may be applied to other applications involving time series data for pattern recognition, analysis and visualisation.

# Chapter 6

# Applications

The last three chapters presented a framework for video sequences alignment with three stages involving video representation, features coding and manifold embedding. The robust video alignment technique can be applied in various fields.

This chapter describes three applications that have been developed to evaluate the presented framework in the video similarity domain. Each section defines one application with a description of the task, overview of the presented approach, dataset used in the experiment and finally the experiment results.

## 6.1 Introduction

This thesis presented a three-stage framework for aligning spatio-temporal information in video streams. The work started with the extraction of interest points from the spatial plane ($xy$) and the temporal planes ($xt$ and $yt$). Then a coding scheme was applied to project each feature into a local coordinate representation produced by a pooling technique. Finally a manifold learning algorithm with spatio-temporal constraints was defined to embed a video clip in a lower-dimensional space while preserving the intrinsic geometry of the data.

Building up the framework from stage one to stage three was combined with an evaluation phase. Three experiments with two tasks were conducted. The first and the second stages (space-time extension of SIFT and modified LLC) were evaluated using the human action classification task. The third stage (manifold embedding) was evaluated using the video clip alignment task. All the experiment results showed that the approaches attained a significant improvement on the defined task over the alternatives.

This chapter further continues evaluation of the entire framework in the video similarity domain using more challenging datasets. To this end, three different applications were defined and developed. A comparison between the three applications is presented in Table 6.1 and summarised below.

1. **Video Clip Retrieval:** The first application is a video clip retrieval based on graph matching. Given a video stream, the task was to measure its similarity to a set of videos and retrieve an unordered list of the most relevant. The query and retrieved videos do not contain any identical content, but they may share similar objects, locations or activities.

   Graphs have drawn attention recently as an effective tool for representation in various computer vision applications. This concept was adopted and the problem of comparing two video clips was reformulated as a graph matching task – *i.e.* by finding the consistent relation between two sets of features through their graphs. More details are presented in Section 6.2.

2. **Video Clip Ranking:** The second application is a video clip ranking task. Given a video stream with repetitive contents, the problem to be addressed is how to define the relation between nearly-repetitive sequences in a low-dimensional space. The approach explores the relationship between the data points and estimates the relevance between the query and remaining retakes in the database. The purpose of the experiment was to create a ranking list of the retakes based on a computed numeric score measuring how each video clip in the database matches the query, and then rank the retrieved clips based on the score. Unlike the video retrieval task, the order of the retrieved clips is important in this task. More details are presented in Section 6.3.

3. **Instance Search Based on Video Queries:** The third application involves searching for a recognizable topic in a query clips within a video streams. It is inspired from the TRECVID instance search task, which gives a collection of clips and a set of query images and where the task is to return the clips related to these query images. The original task was extended to include looking for a segment of video clip containing specific person or location or object as the query video, then retrieving a ranked list of the relevant video clips. The first two applications measure the similarity between a pair of video clips regardless of their contents; however, this third task is more accurate in defining the retrieved

| Properties | Retrieval task | Ranking task | Instance search (INS) task |
|---|---|---|---|
| Dataset | UCF YouTube Action divided into 11 actions | Rushes collection divided into scenes, each contains multiple retakes | Flickr contains 3 entities, each contains 3 recognisable topic |
| Task | Retrieve clips with similar actions as the query | Retrieve all the retakes for the same scene as the query clip, in a ranked list | Retrieve all the clips that contain a recognisable entity defined by the query, in a ranked list |
| Similarity | Either relevant or not based on the similarity | Calculate a similarity score for all the clips | Search for a recognisable object and retrieve the relevant clips, then calculate score and rank |
| Input | One of the actions | A take for one of the scenes | One of the recognisable topics |
| Output | Average precision and recall for each action | Average normalised modified retrieval rank for each query | Similarity score for each query |

Table 6.1: The difference between the three applications developed to evaluate the video sequences alignment framework.

results. Note that it is not a classification task since it searched for specific object of person or location defined later with "topic". More details are presented in Section 6.4.

This task is to give confidence that the developed framework can be applied to more complex applications. The problem was formulated as manifold matching, *i.e.* measuring the similarity between multiple manifolds, each of which represents a video clip. This is inspired by the good results achieved using manifold-to-manifold distance measurements in the face recognition field. Given a query clip the approach explores the relationship between the data points and estimates the relevance between the query and remaining clips in the database.

## 6.2 Video Clip Retrieval

Content-based video-processing has been the centre of attention among researchers in recent years. A wide range of applications rely on accurate representation of visual features so that relevant clips can be identified efficiently [Chen and Chua (2001)]. Although text is the most commonly used form for presenting contents when retrieving information, text alone may not be sufficient for presenting videos because visual

information can be interpreted broadly. Defining vital contents in describing the video content can be subject to human's perception, *i.e.* different users may describe the same scene differently depending on their selection of the visual parts. In addition, most video data are annotated manually by previewing the video, which delivers unsuitable choices of keywords, incomplete sentences which are subject to the annotator's perception, thus affecting the performance of the searching and retrieval. Hence alternative representations of visual contents can be explored.

Video retrieval has two fundamental issues: how to present video contents and how to measure similarity. Graphs have drawn attention recently as an effective tool for representation in various computer vision applications [Chen and Chua (2001)]. Regions of interest in images or videos can be presented as a collection of nodes and edges in a graph. The problem of comparing video clips can then be reformulated as finding the consistent relation between two sets of features through their graphs. Two graphs are aligned by matching their nodes in a way that conserves the edges of the matched nodes [Zaslavskiy et al. (2010)].

Classically, one-to-one methods were considered for most of the graph matching problems [Gibbons (1985)]. Each vertex in the first graph is mapped to only one vertex in the other graph, and vice-versa. For instance, Cour et al. (2007) presented a spectral with affine constraint (SMAC) by introducing affine constraints to the one-to-one matching constraints. This type of matching is too restrictive and cannot be applied to video-processing applications that require matching a set of vertices in one graph to another set of vertices in the first graph. A number of studies have been conducted to address the graph matching problem. Zaslavskiy et al. (2010) proposed a graph matching with continuous relaxation (GM/CR), which is complex approach to detecting many-to-many correspondences between graphs. Although their solution involved discrete optimisation, it is not very suitable for a graph matching problem. Zhou and de la Torre (2012) proposed the factorised graph matching (FGM) algorithm. They factorised the affinity matrix into four smaller matrices: a source and a target graph's incidence matrices, a node and an edge affinity matrix. Therefore a large memory is required to encode the whole affinity matrix without an approximation.

This section presents a novel video retrieval framework based on graph matching. The main focus in this thesis is a video sequences alignment and similarity, which was covered by an evaluation experiment in Chapter 5. To verify the framework within the same domain, the video clip retrieval was chosen as an application. Using the three-stage approach developed in the previous three chapters, the similarity measurement

between a pair of clips is formulated as a graph matching problem in two phases. Firstly, a graph-based representation is defined with a spatio-temporal neighbourhood. Second, the matching problem is converted to a task of measuring the distance between their graphs using the Earth Mover's Distance (EMD) [Rubner et al. (2000)], which is a similarity measurement commonly used in image retrieval and shape matching, measuring the minimum amount of work to transform one representation into the other.

The presented application involves the following points:

*Representation:* A high-dimensional representation is mapped to a spatio-temporal graph, where nodes represent frames and edges represent the temporal order. The graph-based representation is generated using the spatio-temporal neighbourhood graph developed in Chapter 5 as part of the manifold learning. Each node represents a time instance of a video frame sequence, while edges hold the spatio-temporal similarity between two nodes.

*Matching:* The clip similarity is measured using the distance between their graphs. The EMD finds the minimum cost by solving the transportation problem.

*Retrieval:* The unsupervised approach to video clip retrieval, which does not require prior information or training, is presented. Space-time coherence embedded in video sequences is described, then a many-to-many graph mapping technique defines the similarity between clips.

### 6.2.1 The task

Information retrieval methods aim to obtain, from a dataset, the relevant data to a query defined by the user. In this application, the task is defined as: given a query clip represent a human action category, retrieve a set of video clips that contain the same action category from the dataset.

### 6.2.2 The Approach

Given two video clips, the approach is formulated as a graph matching problem in two stages – graph-based video representation (Section 6.2.2.1), followed by similarity measurement between the two clips (Section 6.2.2.2). The following notations is used in the remainder of this section: Let $X = \{x_1, \ldots, x_M\} \in \mathbb{R}^{M \times D}$ denote a query clip

with $M$ frames and $D$ dimensions, where $x_i$ $(i = 1, \ldots, M)$ represents a frame in $X$. Let $Y_k = \{y_1, \ldots, y_N\} \in \mathbb{R}^{N \times D}$ be the $k$-th test clip with $N$ frames and $D$ dimensions, where $y_j$ $(j = 1, \ldots, N)$ represents a frame in $Y_k$.

#### 6.2.2.1 Graph-based Construction

The structure in the high-dimensional space is transferred to a spatio-temporal distance graph with nodes representing frame instances in a query $X$ connecting to the related nodes in a test clip $Y_k$. The method reconstructs the frames order based on their spatio-temporal relationship and recalculates distances along them to ensure the shortest distance.

The procedure is presented with details in Chapter 5, shown in Figure 6.1 and summarised below:

1. Initially a distance matrix $Dist_k = \{dist_{ij}\}$ between two videos $X$ and $Y_k$ is derived, where $dist_{ij}$ is the cost, or the distance, between two frames $x_i$ and $y_j$, calculated by:

$$dist_{ij} = \left( \sum_{d=1}^{D} \|x_{id} - y_{jd}\|^2 \right)^{\frac{1}{2}} \tag{6.1}$$

   where $\| \cdot \|^2$ is the $\ell^2$ norm.

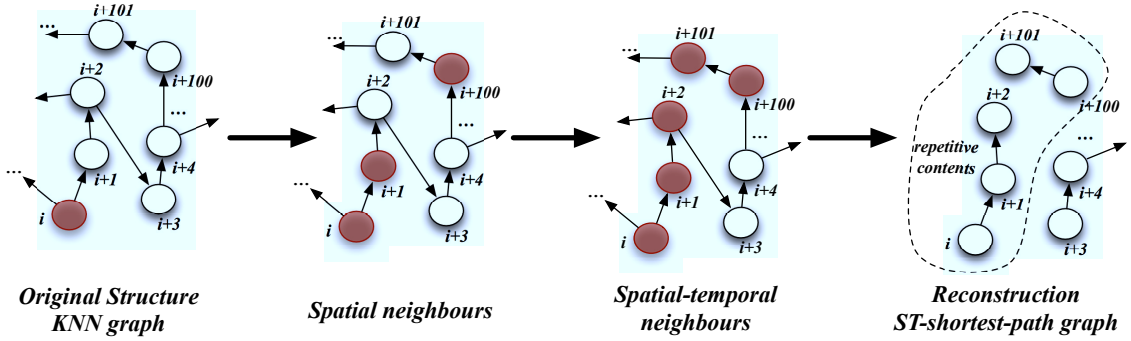



Figure 6.1: The construction of the spatio-temporal distance graph. The initial graph is firstly derived for $x_i$ using the kNN method. Spatial and temporal neighbours are then determined, and the geodesic distance is recalculated by finding the shortest-path using the Dijkstra algorithm. The clusters for repeated structures emerge in the graph.

2. For each frame instance $x_i$ $(i = 1, \ldots, M)$ three sets of neighbours are defined as follows:

- The $L$ frames, with shortest distances $dist_{ij}$ ($j = 1, \ldots, N$), *i.e.* the closest to $x_i$, are connected as spatial neighbours ($sn_{x_i}$).
- Another $L$ frames, chronologically ordered around $x_i$ (as defined by the frames order in the video sequence), are set as temporal neighbours ($tn_{x_i}$).
- Then ($tn_{sn_{x_i}}$) is constructed from temporal neighbours of spatial neighbours.

3. The union of the two neighbour sets; $sn_{x_i}$ and $tn_{sn_{x_i}}$ produces the spatio-temporal neighbours ($stn_{x_i}$) for frame $x_i$.

4. Dijkstra's algorithm is then applied on the constructed neighbourhood graph to find the shortest distances between nodes [van der Maaten et al. (2008)]. This forms a spatio-temporal graph $G_k = \{V_k, E_k\}$ of pairwise geodesic distances, with $V_k = X \cup Y_k$ as the nodes set for both $X$ and $Y_k$, and $E_k = \{\omega_{ij}\}$ as the edge set between each $x_i \in X$ and $y_j \in Y$, and the value of $\omega_{ij}$ represents the distance between two the frames $x_i$ and $y_j$ defined by the shortest path in the graph $G_k$.

#### 6.2.2.2 Graph Matching based on the Earth Mover's Distance

The EMD, initially designed for the image retrieval task [Rubner et al. (2000)], is applied here to define the optimal match between two video clips. EMD has multiple advantages over the other distances measurement including it is ability to measure the distance between variable-size structures and to be applied on different type of representations such as histograms or graphs. The approach assumes that the similarity is modelled by a weighted graph, and the EMD determines the minimum cost of transformation within the graph. It involves solving the well-known *transportation problem* by determining the minimum amount of 'work' required to achieve the transformation. In other words, it seeks to achieve the minimum transactions from a set of nodes that represents frames from the query video to another set of nodes that represents frames from the test video. The name illustrates its concept well; the cost of moving earth from one point to another depends on the distance between the two points and the cost of the transport.

Given two video clips $X$ with $M$ frames and $Y_k$ with $N$ frames, the first step is to construct the spatio-temporal graph $G_k$ (as presented in Section 6.2.2.1) with a set of

nodes $V_k$ with $v_{ij}$ indicating a path from $x_i$ and $y_j$ as shown in Figure 6.2(a), and a cost or weights matrix $E_k$ with $\omega_{ij}$ the cost of moving from frame $x_i$ to $y_j$.

1. The first task is to find the optimal flow $F = \{f_{ij}\}$, where $f_{ij}$ is the transition from a node $x_i$ to the node $y_j$, that minimises the cost function

$$\text{cost}(x_i, y_j) = \sum_{i=1}^{M}\sum_{j=1}^{N}\omega_{ij}f_{ij}, \tag{6.2}$$

   where $F = f_{ij}$ is known as the feasible flow that ensures the non-negativity of the flows and satisfies the following constraints:

   (a) $f_{ij} \geq 0,\ i = 1, \ldots, M,\ j = 1, \ldots, N$ to limit the transition from frame in $X$ to frame in $Y_k$ and not vice versa;

   (b) $\sum_{j=1}^{N} f_{ij} \leq \omega_{x_i},\ i = 1, \ldots, M$ to limit the number of transitions from frames in $X$ to their weights;

   (c) $\sum_{i=1}^{M} f_{ij} \leq \omega_{y_j},\ j = 1, \ldots, N$ to limit the number of transitions to frames in $Y_k$ to be no more than their weights;

   (d) Finally $\sum_{i=1}^{M}\sum_{j=1}^{N} f_{ij} = \min\left(\sum_{i=1}^{M}\omega_{x_i}, \sum_{j=1}^{N}\omega_{y_j}\right)$ to allow the maximum number of transitions between frames.

2. Once this optimal flow $F$ with minimum transportation or flows (as shown in Figure 6.2(b)) is found, the EMD is calculated as the optimal flow computed in Equation 6.2 normalised by the total flow:

$$\text{EMD}(X, Y_k) = \frac{\text{cost}(X, Y_k)}{\sum_{i=1}^{M}\sum_{j=1}^{N} f_{ij}} = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}\omega_{ij}f_{ij}}{\sum_{i=1}^{M}\sum_{j=1}^{N} f_{ij}}. \tag{6.3}$$

   The EMD searches for the feasible path having the minimum cost. A feasible path starts from a vertex in $X$, with a positive flow cost or weight, ends at a vertex in $Y$ with positive flow capacity, *i.e.* entering an intermediate node in $V$ does not affect the total cost of the flow, which is computed as the incoming flow minus the outgoing flow. For instance, Figure 6.2(c) shows a feasible path highlighted by dashed lines which starts at $x_1$, passes $y_2$ and $x_2$, and finally reaches $y_3$.

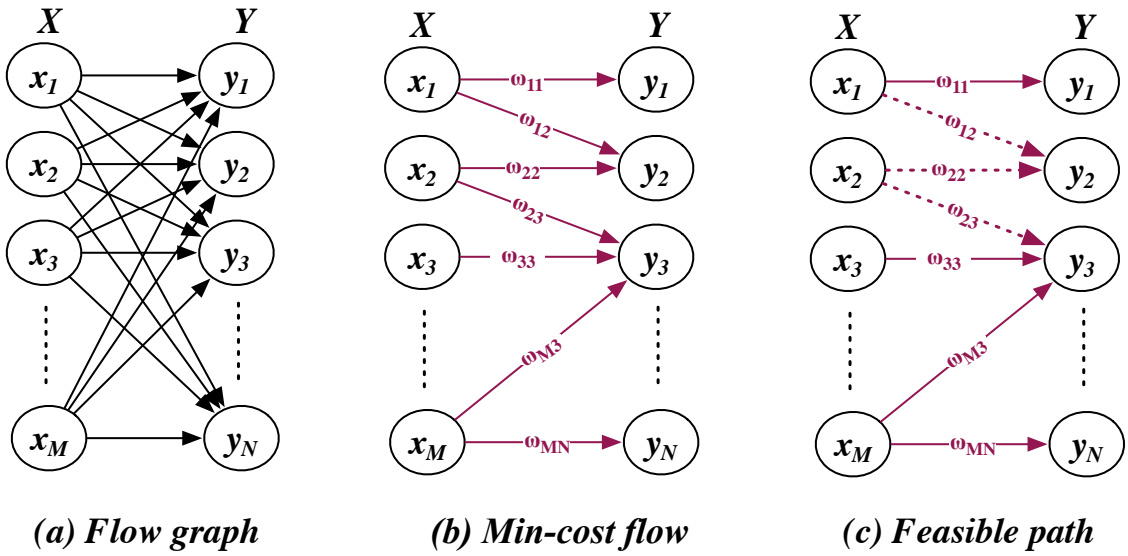



Figure 6.2: EMD computation involves constructing three graphs. (a) The first one is the weighted graph with the shortest paths between the nodes. (b) The second one is the minimum transportation or flows. (c) The last one is the feasible paths with minimum cost.

3. Finally, the similarity between two clips $X$ and $Y_k$ is calculated as:.

$$\text{SIM}(X, Y_k) = 1 - \text{EMD}(X, Y_k). \tag{6.4}$$

Note that $\text{SIM}(X, Y_k)$ ranges between 0 and 1, where $X$ and $Y_k$ are very similar when the value is close to 1.

### 6.2.3 Experiments

The spatio-temporal approach to graph matching was tested using the UCF YouTube action (UCF11) dataset [Liu et al. (2009b)], one of the set of publicly available datasets introduced in Chapter 2 (Section 2.4). There is no specific dataset publicly available for this task. Some of the related work created a small one to serve their goal, however, none of these datasets are publicly available. Therefore, the human action classification datasets were the most suitable one for the retrieval task. The UCF11 consists of 1,600 videos with $c = 11$ action categories. Each category contains 25 scenes with at least four clips for each scene. UCF11 was the most interested and sufficient dataset due to the short duration of the clips, the categorization that helped to define the related

video for each query clip and the complexity of the scenes that were collected from YouTube. For this experiment, 50 clips were randomly chosen to represent each action, two from each scene, totalling 550 clips from the dataset. From the 550 videos, $Nq = 5$ query clips were randomly selected for each action, making 55 videos in the query set.

For the local features detector the ST-SIFT detector as presented in Chapter 3 was applied, combined with the extended LLC introduced in Chapter 4. The initial number of frames to be spatially connected in the manifold appeared dependent on the clip length and was selected manually.

#### 6.2.3.1 Evaluation Schema

The spatio-temporal graph matching with the EMD (GM/EMD) presented in this section was compared with the SMAC [Cour et al. (2007)], the FGM [Zhou and de la Torre (2012)] and the GM/CR [Zaslavskiy et al. (2010)] . As introduced in Section 6.2, SMAC, FGM and GM/CR are all state-of-the-art techniques in graph matching. The criteria of choosing these techniques were firstly to compare with the state-of-the-art, secondly the techniques with publicly available codes to apply them on the video data since all of published works have been employed for image-processing only and finally to measure the performance of the many-to-many against the one-to-one graph matching techniques. To implement these approaches used for comparison, the publicly available code provided by Zhou and de la Torre for the FGM1[1] and by Zaslavskiy for the GM/CR2[2], and the open source Matlab package 'MatlabBGL' for the SMAC3[3] were also used.

Similar to Chapter 5, because the relevant videos for each query clip were pre-defined, the most suitable performance measurement of the four approaches were the precision and recall. The aim was to maximise the number of correctly predicted classes while minimising the number of false positives. The assumption was that video clips from the same action classes would be more similar than those from other classes. Given a query clip $X_i$ from one of five queries $Nq = 5$ $(i = 1, \ldots, Nq)$ belonging to one of the 11 classes $c = 11$ $(j = 1, \ldots, c)$ in the dataset, the following values are firstly defined:

- $TP_i$: number of detected clips that belong to the action class $j$ as query $X_i$

[1] `http:/www.f-zhou.com/gm_code.html`
[2] `http:/cbio.ensmp.fr/graphm/mtmgm.html`
[3] `http://www.timotheecour.com/software/graph_matching/graph_matching.html`

- $FP_i$: number of detected clips that do not belong to the action class $j$ as query $X_i$
- $TN_i$: number of undetected clips that do not belong to the action class $j$ as query $X_i$
- $FN_i$: number of undetected clips that belong to the action class $j$ as query $X_i$.

The average precision and recall, over the number of queries, are then calculated:

$$AP_j = \frac{1}{Nq}\sum_{i=1}^{Nq}\frac{TP_i}{TP_i + TF_i} \tag{6.5}$$

$$AC_j = \frac{1}{Nq}\sum_{i=1}^{Nq}\frac{TP_i}{TP_i + TN_i}. \tag{6.6}$$

Then, using the values from Equation 6.5 and Equation 6.6, the overall average precision and recall over the number of classes in the dataset are calculated:

$$overall\ avrage\ precision = \frac{1}{c}\sum_{j=1}^{c} AP_j \tag{6.7}$$

$$overall\ avrage\ recall = \frac{1}{c}\sum_{j=1}^{c} AC_j. \tag{6.8}$$

#### 6.2.3.2 Results and Analysis

Table 6.2 presents the video retrieval performance using average precision and recall, showing that the GM/EMD outperformed the other techniques. The following observations can be made.

- The GM/EMD proved its ability to represent events in real video sequences with large variations in camera motion, object appearance and pose, scale, viewpoint, cluttered background, illumination conditions, *etc.*. It was able to discover the similarity of clips from the same action category and the dissimilarity between clips that shared some common motions but were from different categories (e.g. 'tennis swinging' and 'golf swinging').
- The EMD has many interesting properties – to some extent it imitates the human perception of texture similarities. It allows partial matches and can be applied to general variable-size signatures.

| Query category | GM/EMD | | SMAC | | FGM | | GM/CR | |
|---|---|---|---|---|---|---|---|---|
| | AP | AR | AP | AR | AP | AR | AP | AR |
| basketball shooting | 0.801 | 0.712 | 0.375 | 0.299 | 0.410 | 0.527 | 0.629 | 0.514 |
| cycling | 0.723 | 0.650 | 0.429 | 0.438 | 0.526 | 0.491 | 0.617 | 0.593 |
| diving | 0.622 | 0.600 | 0.250 | 0.461 | 0.378 | 0.483 | 0.522 | 0.459 |
| trampoline jumping | 0.684 | 0.698 | 0.382 | 0.323 | 0.473 | 0.400 | 0.450 | 0.409 |
| golf swinging | 0.733 | 0.528 | 0.333 | 0.255 | 0.442 | 0.307 | 0.603 | 0.477 |
| horse riding | 0.898 | 0.676 | 0.500 | 0.409 | 0.617 | 0.496 | 0.765 | 0.530 |
| soccer juggling | 0.610 | 0.758 | 0.155 | 0.333 | 0.329 | 0.485 | 0.498 | 0.570 |
| volleyball spiking | 0.790 | 0.754 | 0.380 | 0.399 | 0.501 | 0.489 | 0.588 | 0.565 |
| swinging (by children) | 0.821 | 0.692 | 0.442 | 0.220 | 0.493 | 0.411 | 0.592 | 0.504 |
| walking with dogs | 0.875 | 0.837 | 0.562 | 0.491 | 0.619 | 0.500 | 0.694 | 0.619 |
| tennis swinging | 0.661 | 0.587 | 0.444 | 0.206 | 0.591 | 0.396 | 0.537 | 0.426 |
| Overall average | 0.747 | 0.681 | 0.387 | 0.348 | 0.489 | 0.453 | 0.590 | 0.515 |

Table 6.2: The video clip retrieval task: the average precision (AP) and recall (AR) using the UCF11 dataset. The spatio-temporal graph matching with the EMD (GM/EMD) was presented along with the spectral graph matching with affine constraint (SMAC), the factorised graph matching (FGM) and the graph matching with continuous relaxation (GM/CR).

- The GM/CR approach [Zaslavskiy et al. (2010)] achieved better results than the FGM [Zhou and de la Torre (2012)]. The FGM uses the pairwise relation between feature points with unary information. This relationship is rotation-invariant but not scale-invariant nor affine-invariant.

- Many-to-many mapping approaches (GM/EMD, FGM and GM/CR) performed better than the one-to-one approach (SMAC). One-to-one mapping was too restrictive, while many-to-many mapping allowed flexible matches between the vertices of two graphs. With a many-to-many mapping, object parts can be represented by multiple vertices given noise or various view points. This situation could not be handled by one-to-one matching.

- The SMAC [Cour et al. (2007)] presented the lowest performance by enforcing the one-to-one mapping over video clips. One-to-one mapping does not work effectively when clip contents are similar but not identical because one frame can only be mapped to one corresponding frame.

- The 'basketball shooting', 'trampoline jumping', 'soccer juggling' and 'volleyball spiking' are easily mixed as they all involved 'jumping' motions. The 'cycling' and 'horse riding' categories had similar camera motion. All the 'swinging'

actions shared a common motion. Some actions were performed with objects such as a horse, a bike or a dog. Despite all this, the GM/EMD was able to match and retrieve the relevant clips in many cases.

Figure 6.3 presents retrieved clip samples when a 'basketball shooting' action was given as a query. All the retrieved clips contained the same action, and they were all an indoor scene that was similar to the query. It illustrates how graph representation and matching over the spatial and temporal features improves the video retrieval.

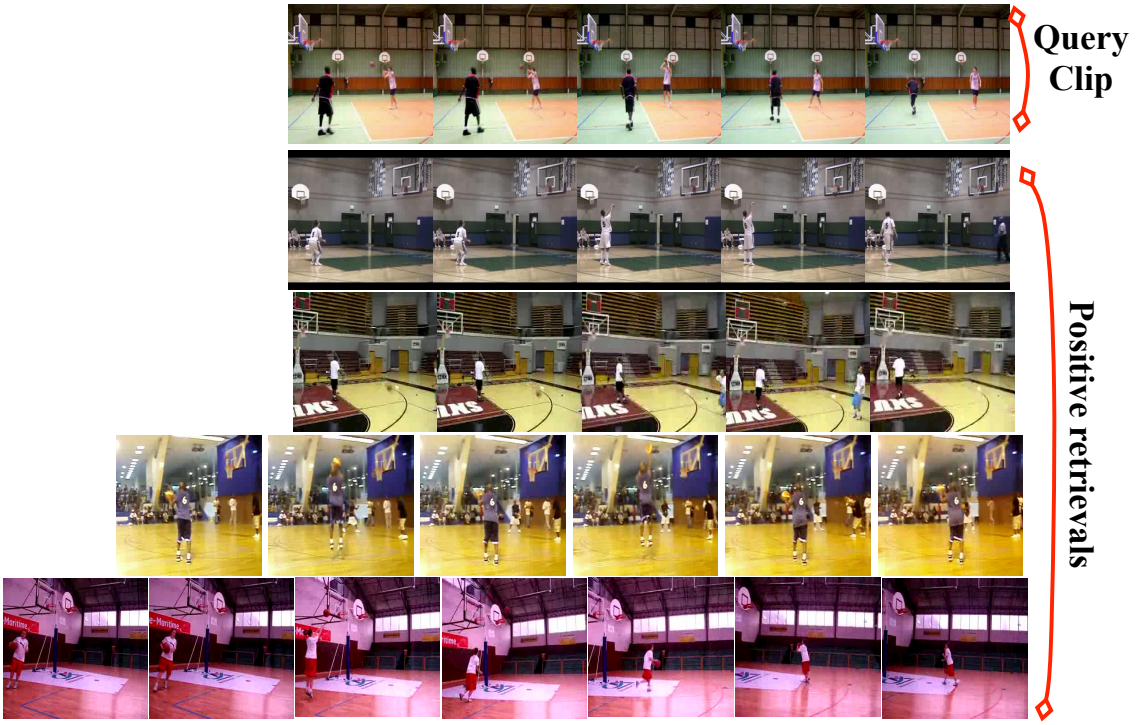



Figure 6.3: The retrieved clip samples from a 'basketball shooting' action query. The top row is the query clip as one of the clips contain the 'basketball shooting' action. Followed by four rows contain positive retrievals that belong to the same action class as the query.

## 6.3 Video Clip Ranking

Analysis of nearly-repetitive video contents has attracted the attention of numerous researchers due to its participating in various fields including news, films, TV programmes and meeting records. It is a challenging task because of the multiple

geometric and photometric transformations caused by different camera angles, digitalisation and editing steps. Rushes video is one example of nearly-repetitive sequences, where the original material is transformed into nearly, but not exactly, identical contents. It is a collection consisting of raw footage and used to produce, for example, TV programmes [Over et al. (2008)].

Unlike many other video datasets, rushes are unconventional, containing additional contents such as clapper boards, colour bars and empty white shots. They also contain repetitive contents, or multiple retakes of the same scene, caused by, for example, actors' mistakes or technical failures. Although contents are nearly repetitive in rushes video, they may not be totally identical duplicates, sometimes causing inconsistency between retakes [Joly et al. (2007)]. Occasionally some parts of the original sequence may be dropped, or extra information may be added at various places, resulting in retakes of the same scene with unequal lengths. Additional transformations, produced by several view settings and multiple camera angles, can occur at various levels from low to high, causing further difficulty in identifying the duplicates.

Most of the previous research on video analysis has been focused on edited videos that are highly structured, such as news and sports. In contrast, rushes video contains unstructured and redundant content, which makes it more suitable and challenging for the retrieval and ranking tasks. It is rich with repetitive retakes containing repeated but not copied contents; at each retake changes are observed to a various extent. It is also important to note that it is relatively simple to define repetitions in rushes; thus they are ideal for a proof of concept for the developed application in this section. Additionally, it was relatively easy for a person to identify repeated contents; thus construction of ground truth had become a simpler task for this dataset than others.

This section presents a framework to rank the nearly-repetitive contents in rushes video. The Section 6.2.2 presented a video retrieval application for evaluating the work. One step forward within the video sequence alignment and similarity domain, which is the focus of this thesis, is the video clip ranking. It analyses the video by characterising the spatio-temporal information embedded in a frame sequence. STG-Isomap developed in Chapter 5 is used as spatio-temporal graph-based manifold embedding to capture the correlations between repetitive scenes. The spatio-temporal intra- and inter-correlations within and between repeated video sequences are defined by applying the ST-LLC introduced in Chapter 4 that is able to detect interest points using the ST-SIFT detector presented in Chapter 3. A series of intrinsic coordinates are then generated in the embedded space using a manifold learning technique.

The presented application involves the following points:

*Representation:* Spatio-temporal intra-correlation is derived as a high-level semantic representation for complex scenes in a video stream. Spatio-temporal interest points are firstly extracted and encoded using fewer codebook bases in the high-dimensional feature space. The correlation within each sequence is derived by constructing a shortest-path graph between the frames using k-nearest neighbour (kNN) searches in both the spatial and the temporal domains.

*Alignment:* STG-Isomap is employed to estimate the synchronisation map for the underlying structure of nearly-repetitive contents in a video stream. An inter-correlation graph between two video sequences reconstructs the underlying structure so that repeated contents can be reorganised, presenting clusters of repetitive scenes.

*Matching:* An unsupervised scheme for ranking, which does not require prior information or pre-processing steps, is presented for multimedia data with repetitions.

### 6.3.1 The task

In this application, the task is to retrieve an ordered list of video clips relevant to the query clip, ranked based on an estimated score. Because of the rushes's nature, containing collections of scenes, each of them contains collections of retakes. The query set contains one take from each scene, while the test set contains the remaining retakes for all the other scenes including the other retakes of the query clip. The task can be restated as: given a query clip represents a take for one of the scenes, retrieve all the retakes (for that specific scene) ordered from the most similar to the least similar based on the calculated score. The similarity was determined based on the ground truth that defined the set of retakes for each scene; however, the similarity between the retakes of the same scene was determined based on the calculated score.

### 6.3.2 The Approach

Given a video stream with repetitive contents, the problem to be addressed is how to define the relation between nearly-repetitive sequences in a low-dimensional space. The approach consists of two stages. First, spatio-temporal intra-correlation between

the frames of each video sequence is captured in a high-dimensional space using space-time invariant interest points detection and coding approaches (Section 6.3.2.1). Second, a manifold representation is computed for each video scene to map the high-dimensional representation to the embedded space (Section 6.3.2.2). At this stage the inter-correlation is computed between multiple video sequences using the spatio-temporal kNN graph. A spatio-temporal Isomap technique is used to generate the intrinsic coordinates for each sequence in the manifold. Generated coordinates are chronologically ordered according to the spatio-temporal similarity and integrated to define a graph for sequence alignment. Figure 6.4 illustrates the entire process of the framework.

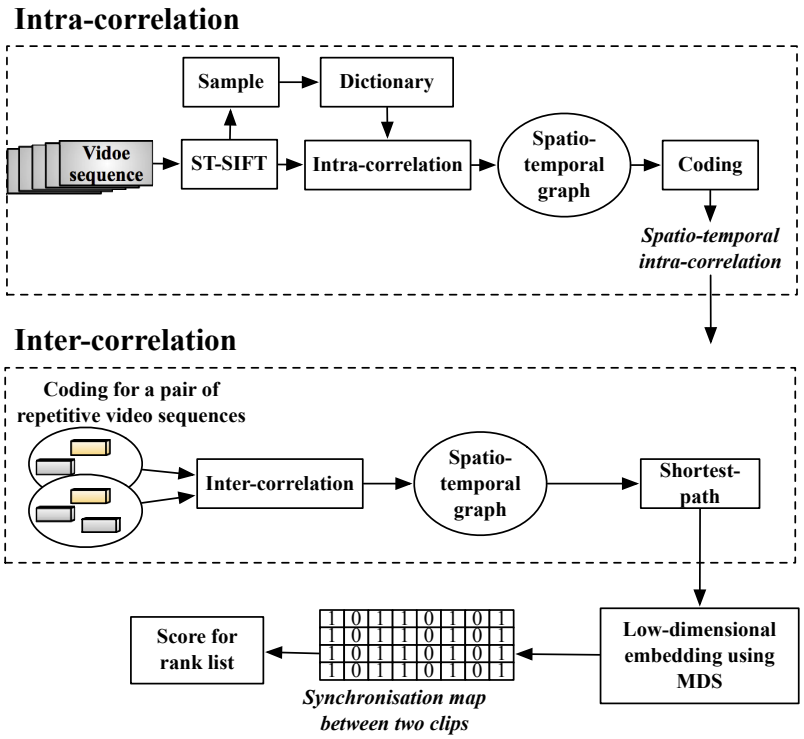



Figure 6.4: Processing steps for nearly-repetitive contents detection. The upper part shows the construction of the intra-correlation between video frames: by constructing a shortest-path graph between the frames using kNN search in both the spatial and the temporal domains. The lower part shows the computation of the inter-correlation between multiple video sequences: constructiing a spatio-temporal neighbourhood graph, calculating the shortest path distance and then mapping then to the lower dimensional space. A synchronisation map between two clips was then calculated and used to rank the repetitive contents.

#### 6.3.2.1 Spatio-temporal Video Representation

The ST-SIFT detector combined with the extended LLC was used to define the spatio-temporal video representation.

**Spatio-temporal SIFT**

The ST-SIFT algorithm, introduced in Chapter 3, is used to identify spatially and temporally invariant interest points from a video stream. To achieve the invariance in both space and time, a spatio-temporal Gaussian and difference of Gaussian (DoG) pyramids are calculated first. Then the points shared between three spatial and temporal planes ($xy$, $xt$ and $yt$) at each scale in the DoG are chosen as interest points.

**Coding with a Shortest-path Graph**

LLC, described in Chapter 4, is a spatial coding scheme extended to space-time domain to project individual descriptors onto their respective local coordinate systems. Given the ST-SIFT feature matrix extracted from a video stream with $N$ entries and $D$ dimensions, *i.e.* $X = \{x_1, \ldots, x_N\} \in \mathbb{R}^{D \times N}$, LLC seeks to define the $S = \{s_1, \ldots, s_N\} \in \mathbb{R}^{D \times N}$ as a set of spatio-temporal codes for $X$.

Intra-correlation within each sequence $S_x$ is derived by firstly constructing a spatio-temporal graph between the descriptors and the codebook, then computing the shortest-path and performing a kNN search and finally solving a constrained least-squares fitting problem.

#### 6.3.2.2 Manifold Learning

Given the feature matrices for two video clips, $X = \{x_1, \ldots, x_N\} \in \mathbb{R}^{D \times N}$ with $N$ entries and $Y = \{y_1, \ldots, y_M\} \in \mathbb{R}^{D \times M}$ with $M$ entries, spatio-temporal codings, $S_x$ and $S_y$, are calculated as defined in Section 6.3.2.1. To estimate the synchronisation map between $X$ and $Y$, STG-Isomap was applied as presented in Chapter 5 to define a new correlation $\delta_\gamma$ from a spatio-temporal correlation $\delta_{xy}$. The procedure is summarised below:

1. Construct the spatio-temporal kNN graph $\delta_{xy}$ between the two intra-correlation matrices $S_x$ and $S_y$. Each node represents a spatio-temporal interest point in one of the sequences, while each edge represents a connection between two events if they are related.

2. Compute the geodesic distances between the nodes in graph $\delta_{xy}$.

3. Define the $K$ spatial neighbours for each frame $x_i$ using the shortest-path.

4. Define the another $K$ temporal neighbours for each frame $x_i$ based on their chronological order.

5. Define the temporal neighbours of the spatial neighbours for further coverage.

6. Construct a union between the two sets from step 3 and step 5 to represent the spatio-temporal neighbours.

7. Construct the inter-correlation by re-calculating the shortest-path between the nodes in graph $\delta_{xy}$, to form a new embedded correlation $\delta_{\gamma}$.

8. Model the manifold embedding as a transformation $T$ of the inter-correlation $\delta_{\gamma}$ into a new embedded space $U$. The optimal distance graph is generated where nodes represent features and edges represent the spatio-temporal relationship between the nodes. Finally, the graph-based representation is mapped to a lower-dimensional space using MDS.

### 6.3.3 Experiments

A video clip ranking experiment was conduced to assess the performance of the nearly-repetitive contents detection framework. From the set of publicly available datasets (introduced in Chapter 2 (Section 2.4)), the NIST TRECVID 2008 BBC rushes video collection was used [Over et al. (2008)]. Ten video sequences were selected, with an approximate total duration of 183 minutes. The rushes videos are the most interested and challenging datasets for video-processing applications. Choosing it for the ranking application was because of its nature. The short duration of the video clips, the high degree of the similarity between the clips since they are repetitions of the same scene, as well as the ability to define the ground truth from the descriptions attached with this data.

The purpose of the experiment was to retrieve multiple similar retakes of the same scene in a ranked list. The ground truth was created for each video following the same procedure presented in Chapter 5 Section 5.4.1. The defined positions for ten videos, totalling 64 scenes with 189 retakes, were used as the ground truth. Table 6.3 provides a further breakdown of the dataset.

The video representation was created as by firstly extracting the ST-SIFT interest points from the segmented frames as presented in Chapter 3. Then the spatio-temporal

| video id | duration (min:sec) | #scenes | #retakes (#retakes/scene) |
|---|---|---|---|
| *MS206290* | 21:03 | 7 | 23 (2,3,5,5,2,2,4) |
| *MS206370* | 12:30 | 7 | 17 (2,2,2,2,3,2,4) |
| *MS215830* | 14:55 | 5 | 14 (3,3,3,2,3) |
| *MS220770* | 18:50 | 4 | 12 (3,2,4,3) |
| *MS221020* | 29:18 | 11 | 38 (2,3,5,2,4,4,2,3,5,6,2) |
| *MS221050* | 16:40 | 4 | 11 (3,4,2,2) |
| *MRS159318* | 18:09 | 5 | 15 (4,3,3,3,2) |
| *MRS146579* | 19:43 | 8 | 26 (2,4,4,5,3,3,2,3) |
| *MRS044499* | 12:42 | 4 | 8 (2,2,2,2) |
| *MRS150072* | 21:40 | 9 | 25 (3,5,2,2,3,2,3,3,2) |

Table 6.3: Dataset created with ten video clips from the TRECVID 2008 BBC rushes video collection [Over et al. (2008)]. Each video stream consists of a number of scenes, each with multiple retakes.

regions around the interest points were described by the 3D-HOG descriptors [Scovanner et al. (2007)]. This was followed by applying the extended version of LLC introduced in Chapter 4. The spatio-temporal codes were defined for each spatio-temporal sub-region and pooled together using the multi-scale max pooling to create the mid-level representation. The experiment used $4 \times 4$, $2 \times 2$ and $1 \times 1$ sub-regions. The pooled features were then concatenated and normalised using the $\ell^2$-norm.

Given a query clip, a synchronisation map was estimated between the query and each clip in the dataset using the manifold embedding. The positive predictive value (PPV, also known as the precision rate) was calculated to drive the proportion of positive test results:

$$PPV = \frac{TP}{TP + FP}, \tag{6.9}$$

where $TP$ is the number of the correctly detected retakes that belong to the same scene, and $FP$ is the number of incorrectly detected retakes that do not belong to the same scene. Finally a ranked list was created with video clips ordered by their PPVs.

#### 6.3.3.1 Evaluation Schema

The average normalised modified retrieval rank (ANMRR) [MPEG video group (1999)] was used for the evaluation. This was the normalised ranking method used by the MPEG group to evaluate the system performance. It is employed in this application

because it considers not only the recall and precision information, but also the rank information among the retrieved videos. Values range between 0 and 1, where values closer to 0 imply that relevant clips were ranked higher in the list. ANMRR was calculated by

$$ANMRR = \frac{1}{Q}\sum_{q=1}^{Q}\frac{MRR(q)}{K + 0.5 - 0.5 \times NG(q)}, \tag{6.10}$$

where

$$MRR(q) = \frac{1}{NG(q)}\left\{\sum_{k=1}^{NG(q)} Rank(k)\right\} - 0.5 - \frac{NG(q)}{2}. \tag{6.11}$$

For queries $q = 1, \ldots, Q$, $NG(q)$ is the number of relevant clips in the database defined by the ground truth. Further, $K = \min\{4NG(q), 2GMT\}$, with $GMT = \max\{NG(k)\}$. $Rank(k)$ of the $k$-th ground truth clip is the position at which this clip was ranked by the PPV score calculated with Equation (6.9). It defines the relevant ranks, *i.e.* if the clip was in the first $K$ retrievals then $Rank(k)$ was kept, otherwise $Rank(k) = K + 1$.

#### 6.3.3.2 Retakes Ranking

The experiment was conducted with $K = 6$ and $Q = 20$ clips (two from each video sequence) chosen randomly, each of which was used as a query. The rest (188 clips) were the pool of candidates; the retakes of the query were treated as the ground truth for relevant clips to be retrieved. To assess the performance, the approach (Framework 1) was compared with two simplified alternatives (Frameworks 2 and 3). The purpose of defining these frameworks was to measure the advantages of the developed extensions in both video representation and manifold embedding stages against the original techniques from the spatial domain.

**Framework 1:** ST-LLC/ STG-Isomap.
A combination of ST-SIFT and LLC coding with the shortest-path graph was used for intra-correlation, which was followed by inter-correlation using STG-Isomap;

**Framework 2:** ST-LLC/ Isomap.
STG-Isomap of Framework 1 was replaced with the conventional spatial Isomap;

| query clip ID | #relevant clips $NG(q)$ | Framework 1 | Framework 2 | Framework 3 |
|---|---|---|---|---|
| 1 | 4 | 0.116 | 0.314 | 0.224 |
| 2 | 3 | 0.020 | 0.211 | 0.317 |
| 3 | 2 | 0.167 | 0.299 | 0.405 |
| 4 | 3 | 0.000 | 0.213 | 0.272 |
| 5 | 2 | 0.000 | 0.296 | 0.288 |
| 6 | 2 | 0.193 | 0.303 | 0.481 |
| 7 | 3 | 0.100 | 0.215 | 0.174 |
| 8 | 2 | 0.000 | 0.090 | 0.120 |
| 9 | 5 | 0.201 | 0.311 | 0.329 |
| 10 | 4 | 0.188 | 0.218 | 0.200 |
| 11 | 3 | 0.099 | 0.266 | 0.107 |
| 12 | 2 | 0.208 | 0.299 | 0.487 |
| 13 | 3 | 0.197 | 0.355 | 0.360 |
| 14 | 2 | 0.151 | 0.210 | 0.267 |
| 15 | 4 | 0.111 | 0.234 | 0.391 |
| 16 | 3 | 0.204 | 0.260 | 0.494 |
| 17 | 1 | 0.088 | 0.279 | 0.166 |
| 18 | 1 | 0.162 | 0.199 | 0.181 |
| 19 | 2 | 0.030 | 0.200 | 0.194 |
| 20 | 4 | 0.000 | 0.293 | 0.390 |
| ANMRR | | 0.112 | 0.253 | 0.292 |

Table 6.4: Video retrieval experiment using nearly-repetitive contents. Two clips were selected from each video: *MS206290* (Clips 1,2), *MS206370* (3,4), *MS215830* (5,6), *MS220770* (7,8), *MS221020* (9,10), *MS221050* (11,12), *MRS159318* (13,14), *MRS146579* (15,16), *MRS044499* (17,18), *MRS150072* (19,20). ANMRR was the average of NMRRs for individual queries.

**Framework 3:** SIFT/ LLC/ STG-Isomap.

ST-SIFT of Framework 1 was replaced with the conventional 2D SIFT and LLC coding with the Euclidean distance graph.

Table 6.4 presents the outcome of the video retrieval experiment. Figures 6.5 and 6.6 show two ranking examples by Framework 1, where the first row was the query clip and highly ranked retakes are shown from the second row.

Table 6.4 shows that the approach presented in this section (Framework 1: ST-LLC/ STG-Isomap) outperformed its simplified alternatives by a fair margin (0.14 and 0.18 absolute against Frameworks 2 and 3). The main reasons were: (1) Spatio-temporal intra-correlation was a multi-region frame-by-frame comparison that was able to capture the similarities between two frames along the time line. This was due to the fact that, in a sense, characteristics and locations of individual objects

were coherently transitioned from one frame to another. ST-LLC was able to detect the significant points in the events and track these points along the video sequence. Further, each descriptor was projected to its local coordinate system to create the video representation. (2) The inter-correlation helped to reconstruct the video sequences in the lower-dimensional space in order to uncover their repetitive contents. The retakes from the same scene were mapped close to each other in the manifold, resulting in clusters of repetitive contents. Pairs of the closest frames from the different retakes would form an alignment.

Framework 2 (ST-LLC/ Isomap) performed better than Framework 3 (SIFT/ LLC/ STG-Isomap) for videos *MS206370* (queries 3 and 4) and *MRS150072* (queries 9 and 10). This was because of the variations in scene setting such as the appearance of dominant colour patterns and the changes of video shooting location within the scene. Such variations had to be captured spatially and tracked temporally, and this was handled by Framework 2 better than Framework 3.

On the other hand, Framework 3 resulted in better ANMRR scores than Framework 2 with indoor scenes such as *MRS044499* (queries 7 and 8). Framework 3 was more able to capture the similarity between multiple retakes spatially. While not as good as Framework 1, the overall performance by Framework 2 with the spatio-temporal intra-correlation was better than Framework 3. Some scenes contained camera or object movement, thus causing discontinuity within the sequence because they did not share sufficient spatial features with their neighbours. Consideration of the temporal relation alleviated this problem.

In the pool of candidates, there were scenes with different scenarios or events that occurred at the same location. Similarity calculation often clustered these scenes together. Another interesting example was the semantic similarities between different scenes that shared same concept (e.g. activities). This example is illustrated in Figure 6.6 from video *MS206290*. Given a query containing a crowd of people, the highest-ranked clip also contained a crowd of people but from a different scene. Although Framework 1 created good ranked lists, such scenarios often caused erroneously high ranking of unrelated scenes.

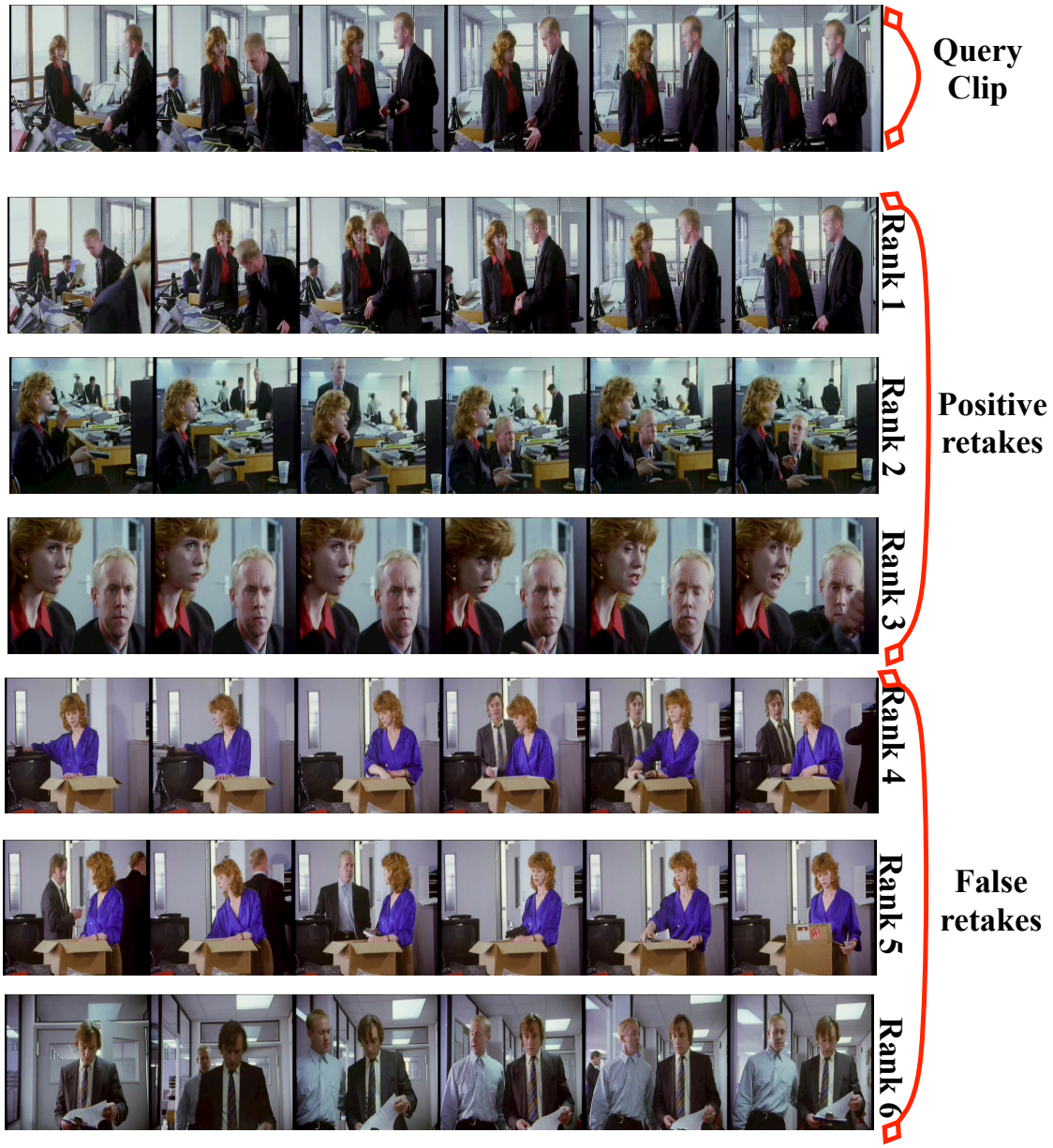



Figure 6.5: The query (top row) was one of four retakes in the last scene of video '*MS206370*'. Three relevant clips were ranked first, second and third out of 188 candidates. After that, the false retakes were produced as illustrated from ranks 4 to 6 (last three rows) that actually belong to the fifth scene.

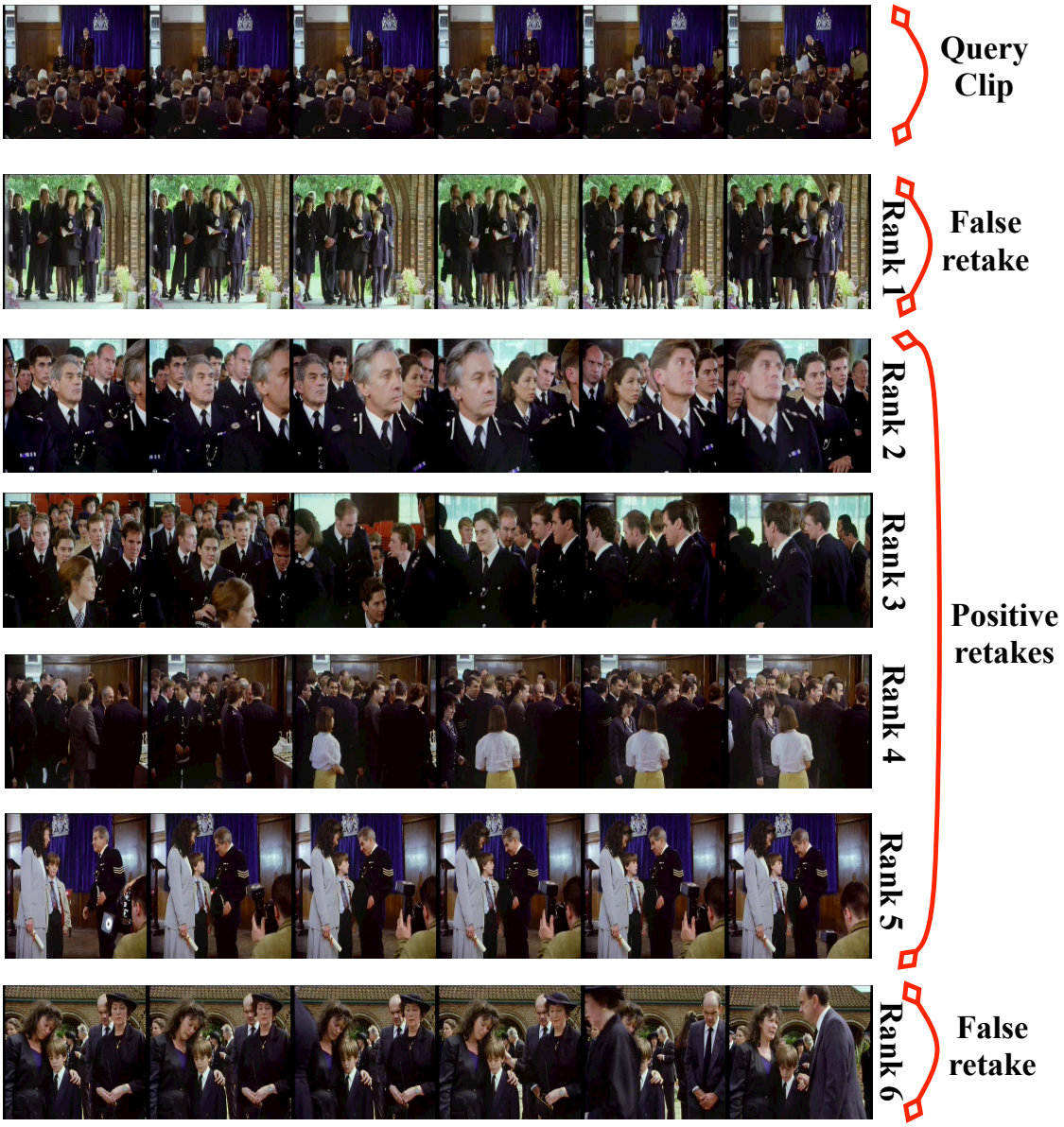



Figure 6.6: The query (top row) was one of five retakes in the third scene of video '*MS206290*', hence there were four relevant clips to be retrieved. The first and sixth rank (the second and last row) were a false rank caused by the semantic similarity between the query and the retake containing a crowd of people. Ranks 2 to 5 were all relevant clips (true positives).

## 6.4 Instance Search Based on Video Queries

Over the past decade, there has been an exponential growth of multimedia, in particular video data such as surveillance, commercial, news and personal videos, available on the network. There is a need to be able to locate important entities such as persons, objects and places in a video segment, given a visual query. To this end the NIST has been organising TRECVID, a series of annual laboratory style evaluations in the video-processing field. Every year challenging tasks with a large collection of video data have been provided for content analysis and information extraction. The recent instance search task was concerned with identification of a specific person, an object or a place in a video data collection, whereby a still image was provided as a query. A typical approach involved extraction of objects of interest, followed by a choice of machine learning algorithms. However their performance was generally at a laboratory level under some conditions including illumination, camera angles and pose variations. Additionally, given the massive scale of video collections, the amount of objects of interest extracted from both training and testing data can be huge, potentially causing a computation problem.

Manifold learning such as Isomap [Tenenbaum et al. (2000)], LLE [Roweis and Saul (2000)] and laplacian eigenmap [Belkin and Niyogi (2003)] were developed for modelling data sets from a single manifold. They were good at finding low-level coordinates that preserved original geometric representation. However, because of their non-linearity embedding, it was hard to apply them directly to new test samples [Wang et al. (2008)]. This narrowed their possible applications to classification and recognition. Recently, to address the issue of non-linearity constraints, several approaches have been proposed such as locality preserving projections (LPP) [He and Niyogi (2003)] and unsupervised discriminant projection (UDP) [Yang et al. (2007)]. These provide a linear mapping with simple computation and achieve good results with the face recognition task.

More recently, from the context of manifold learning, several algorithms have been developed to define the manifold distance measurement. Manifold-manifold distance (MMD) was developed by Wang et al. (2008) to measure the similarity between two sets of facial images. They applied the maximal linear patch (MLP) method to cluster sample images and learn linear subspaces. MMD calculates the average between two types of distances, the exemplar distance and the variation distance. The exemplar distance is the correlation between the orthogonal of two samples, while the varia-

tion distance is the canonical angle between them. Fitzgibbon and Zisserman (2003) represented detected faces by affine subspaces and applied a joint manifold distance (JMD) to measure the distance between subspaces. Their method involves a parameter estimation problem and could fail where the statistical relation is weak between the testing and the training data. In addition a large set of training data is required to approximate the distribution functions. Souvenir and Pless (2005) defined weighted Isomap using the expectation maximisation (EM) type technique to cluster manifolds, which could fail where the clusters were widely separated. All of these works addressed the face recognition problem from a collection of images, where a query was a set of images for the same individual. Their methods depend on training a model for each individual using a large set of samples, and then measuring the similarity between these trained models and the query model.

In this section, the INS task was extended to process video clips as query and retrieved result. The aim of choosing this application was to rise the challenge and give more confidant that the developed framework is able to deal with video content in different situation. In video similarity domain, the INS is considered one of the challenging task and it is been used for competition and testing different frameworks and baselines. The problem was formulated as manifold-to-manifold matching based on their distance in the lower-dimensional space. The idea is to measure the distance between a manifold constructed from a clip in the video collection and a query manifold constructed from a query video. To the best of our knowledge there has been no study of manifold embedding with the video search/retrieval task. Further, this work is novel in that it does not require a template or training data. Instead it analyses the video by characterising the spatio-temporal information embedded in a frame sequence. The STG-Isomap presented in Chapter 5 is firstly applied as a spatio-temporal graph-based manifold embedding, to discover the underlying structure of a video stream. Motivated by the local linear models construction in [Wang et al. (2008)] and the distance measurement in [Moghaddam et al. (2000)], a manifold representation is defined as a set of locally linear models, each of which can be interpreted as a subspace. The manifold matching is then solved by measuring the similarity between a pair of subspaces, each from one of the manifolds. The linearity-constrained hierarchical divisive clustering (L-HDC) algorithm [Wang et al. (2008)] is applied for the video clip manifold to construct the local clusters of the similar video entity. Finally these clusters are used to generate models and subspaces that will be used to solve the matching problem.

The presented application involves the following points:

*Defining the problem:* The video clip search problem is formulated explicitly as measurement of the manifold similarity. A manifold representation and matching method was proposed, with which a non-linear manifold is transformed to a set of locally linear models and the distances between them is used as measurements.

*Representation:* STG-Isomap is applied to estimate the underlying structure of video contents and define a spatio-temporal intra-correlation in a video stream.

*Matching:* A novel method is introduced to extract linear models with locality-constrained from the spatio-temporal representation of a video stream. These models are constructed from spatio-temporal patches that characterise a certain event in the video. Distance measurement between a set of linear models, each of which is depicted by a subspace, is defined where dual-subspace and principal angles are used.

*Evaluation:* An extension is made for the TRECVID instance search task to locate a certain entity within a collection of video clips and retrieve a ranked list of the relevant query clips. The approach was applied to real data, and very good results were achieved.

In spirit, this section bears some resemblance to Fan and Yeung (2006) and Wang et al. (2008) in that they too used a manifold embedding technique for representation and distance measurement for matching. It also shares the idea of using dual-subspace and principal angles as the distance measurement [Fan and Yeung (2006)]. However earlier works were developed mainly for face recognition in the spatial domain and none of them employed a temporal relation. In addition, both works trained a model for each individual face and measured the similarity based on these models. This work, the detail of which is presented in the following sections, is notably different.

### 6.4.1 Levels of Distance Functions

In classification and recognition tasks, objects of interest can be defined at three levels: point (for a single data), subspace (for a model spanned by a set of points), and manifold (for lower-dimensional embedding spanned by a set of subspaces) [Wang et al. (2008)]. This section overviews these three levels, with the various measurements that can be defined between them (shown schematically in Figure 6.7).

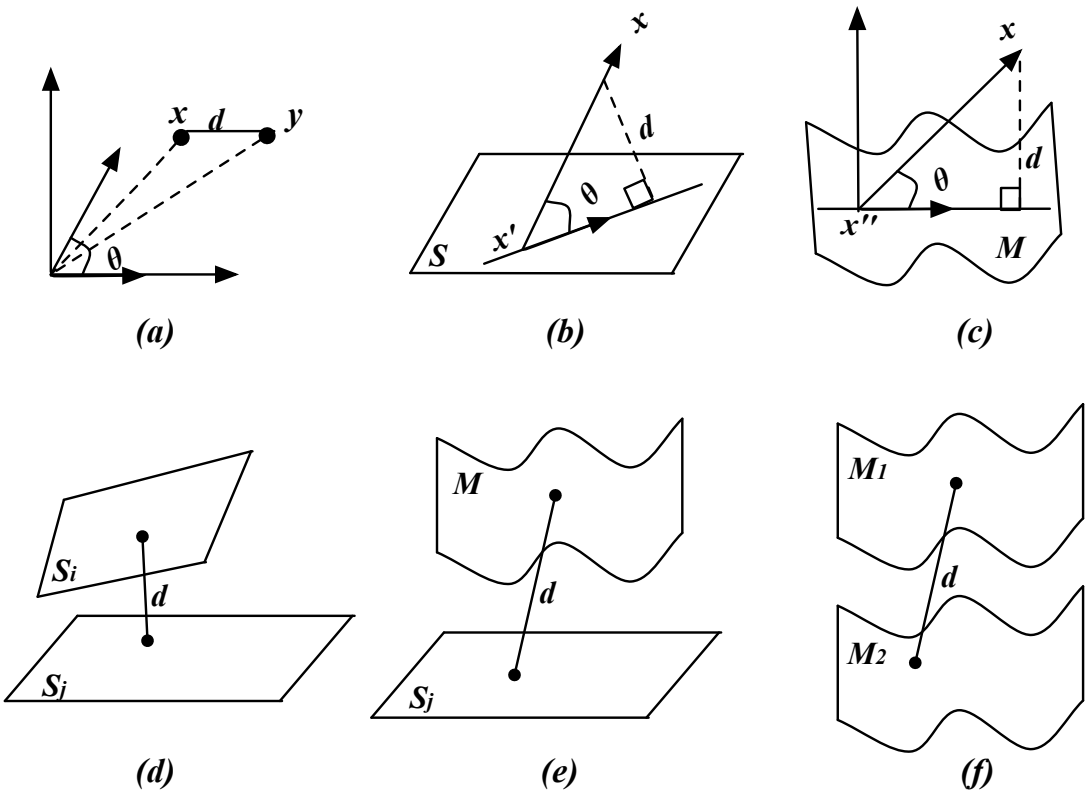



Figure 6.7: Different levels of distances defined over points, subspaces and manifolds: (a) point-point distance; (b) point-subspace distance; (c) point-manifold distance; (d) subspace-subspace distance; (e) subspace-manifold distance; (f) manifold-manifold distance. Figure taken from [Wang et al. (2008)].

The point and the subspace distances have been well covered in the literature, while studies concerned with the manifold distance were relatively rare in the past. Formally, points are referred to by $x$ and $y$, subspaces by $S$ and manifolds by $\mathcal{M}$, and the manifold $\mathcal{M}$ is expressed as a set of $L$ subspaces:

$$\mathcal{M} = \{S_i : i = 1, 2, \ldots, L\} = \{S_1, S_2, \ldots, S_L\}, \tag{6.12}$$

where the $S_i$ is the $i$-th component subspace of the manifold.

#### 6.4.1.1 Point Distances

Three types of distances can be formulated over the points: point-point distance (PPD), point-subspace distance (PSD) and point-manifold distance (PMD).

- **Point-point distance (PPD)**

PPD is defined between two points $x$ and $y$ as $d(x, y)$ and is commonly calculated by the Euclidean distance:

$$d(x, y) = \|x - y\|. \tag{6.13}$$

- **Point-subspace distance (PSD)**

PSD is defined between a point $x$ and a subspace $S$ as $d(x, S)$ and is generally measured by the $L_2$-Hausdroff distance:

$$d(x, S) = \min_{y \in S} \|x - y\| = \|x - x'\|, \tag{6.14}$$

where $x'$ is the projection of the point $x$ in the subspace $S$. The PSD can be expressed as the PPD between the point $x$ and its projection $x'$ in $S$.

- **Point-manifold distance (PMD)**

PMD is defined between a point $x$ and a manifold $\mathcal{M}$ as $d(x, \mathcal{M})$ and it can be calculated as follows:

$$\begin{aligned} d(x, \mathcal{M}) &= \min_{S_i \in \mathcal{M}} d(x, S_i) \\ &= \min_{S_i \in \mathcal{M}} \min_{y \in S_i} \|x - y\| \\ &= \|x - x''\|, \end{aligned} \tag{6.15}$$

where $x''$ is the nearest point to the point $x$ in the manifold $\mathcal{M}$.

#### 6.4.1.2 Subspace Distances

One level above the point representation, two types of distances can be calculated over the subspaces: subspace-subspace distance (SSD) and subspace-manifold distance (SMD).

- **Subspace-subspace distance (SSD)**

SSD is defined between two subspaces $S_i$ and $S_j$ as $d(S_i, S_j)$. The most common measurement for this distance is the principal angle [Bjoerck and Golub (1971)] as this generally achieves a satisfactory performance. Formally it can be defined as:

$$\begin{aligned} d(S_i, S_j) &= \min_{x \in S_i} \|x - S_j\| \\ &= \min_{x \in S_i} \min_{y \in S_j} \|x - y\| \\ &= \|x - x'\|, \end{aligned} \tag{6.16}$$

where $x$ and $x'$ are the closest points between the two subspaces.

- **Subspace-manifold distance (SMD)**

SMD is defined between a subspace $S_i$ and a manifold $\mathcal{M}$ as $d(S_i, \mathcal{M})$. Similar to the PMD, it is expressed as the nearest subspace to $S_i$ in manifold $\mathcal{M}$:

$$d(S_i, \mathcal{M}) = \min_{S_j \in \mathcal{M}} d(S_i, S_j) = \|S_i - S'\|. \tag{6.17}$$

#### 6.4.1.3 Manifold-manifold Distance

Motivated by the fact that a global non-linear manifold preserves the locality linear property anywhere in the manifold, the manifold can be defined by a set of local linear subspaces [Roweis and Saul (2000)]. Thus the distance between a pair of manifolds is converted to the distance between their best suited subspaces. It can be expressed as an extension of the subspace-subspace distance with more complex data variations.

Formally, given a pair of manifolds $\mathcal{M}_1 = \{S_1, \ldots, S_L\}$ with $L$ local linear subspaces $S_i$, and $\mathcal{M}_2 = \{S_1, \ldots, S_N\}$ with $N$ local linear subspaces $S_j$, the distance between them can be defined as $d(\mathcal{M}_1, \mathcal{M}_2)$. Using the local linear models representation in Equation (6.12), the MMD can be expressed as finding the shortest distance between a pair of subspaces from the two manifolds:

$$\begin{aligned} d(\mathcal{M}_1, \mathcal{M}_2) &= \min_{S_i \in \mathcal{M}_1} d(S_i, \mathcal{M}_2) \\ &= \min_{S_i \in \mathcal{M}_1} \min_{S_j \in \mathcal{M}_2} d(S_i, S_j). \end{aligned} \tag{6.18}$$

### 6.4.2 The task

The TRECVID aims to promote progress in various content-based analysis and retrieval tasks using digital video datasets [Over et al. (2008)]. It is a laboratory-style

evaluation that seeks to define different tasks reflecting real world situations or significant parts of such situations. In 2014 NIST evaluated the submitted systems on five different tasks including: semantic indexing, interactive surveillance event detection, multimedia event detection, multimedia event recounting and INS. The INS task is considered the most challenging that aims to locate video clips from dataset that contain an entity with recognisable topic and defined by a query image. A case study would be defined as a user browsing a video dataset and locating a person, place, or object of interest in one video, either known or unknown, and he wants to retrieve more videos containing the same target, but not necessarily in the same context. Example of objects would be an Audi logo, a cigarette, a black taxi *etc.*, while places would be Stonehenge, Pantheon interior, Prague Castle *etc.*, and finally persons would be Stephen Colbert, Barack Obama, Singer *etc.*

Inspired by the TRECVID task, a task was defined to deal with video INS based on a video stream as both test and query data. Given a collection of test video clips and a collection of query video clips containing either a person, an object, or a place entity, the aim was to locate for each query up to a specific number of clips that most likely contain a recognisable instance of the entity (referred to as a topic).

The INS can be considered as ranking or retrieval task but with more constraints:

- The similarity within retrieval and ranking tasks is more general, looking for same actions, same environment *etc.*, and does not consider a specific entity. In contrast, the INS searches for a recognisable entity regardless of the action performed or the environment condition.
  The retrieval task divided the dataset into related or unrelated to the query clip based on their general similarity, *i.e.* without looking for a recognisable object, and the ranking task calculated the similarity score for all the clips in the data set and then created an order list of all clips based on their score values. The INS firstly locates or searches for a recognisable entity in the video clips, then calculates a similarity score and finally retrieves the results in an ordered list.

- The INS task can act as a retrieval task - both divide the dataset into relevant or non-relevant sequences to the query clip; however, the INS searches for a recognisable entity. The INS task can also act as a ranking task - both order the results based on a similarity score; however, the INS orders the relevant set only, while the ranking task orders the whole dataset.

- The nature of the datasets used for the INS and the ranking task are totally

different. The rushes collection used in the ranking task contains sets of scenes where each one has multiple retakes that can be similar or even identical, while the Flickr dataset used in the INS contains different clips with different content, actions, actress *etc.*, and only share the recognisable entities defined by the queries.

### 6.4.3 The Approach

This section first formulates the extended instance search task. Then it describes the video representation stage based on the spatio-temporal graph-based manifold embedding. Finally it introduces the manifold matching algorithm by emphasising its two main steps: local linear models construction and the distance measurement between them.

#### 6.4.3.1 Problem Formulation

Given a query clip with recognisable person, object, or place entity, the aim is to retrieve all clips that most likely contain the same instance of the entity (referred to as a topic). Figure 6.8 briefly illustrates the concept of this application. Formally, a video database with $C$ topics is considered, where each topic $c$ ($c = 1, 2, \ldots, C$) has a set $V_c = \{v_{c,1}, v_{c,2}, \ldots, v_{c,F}\}$ of $F$ videos, where $v_{c,k} = \{v_1, \ldots, v_N\} \in \mathbb{R}^{N \times D}$ is the $k$-th test clip with $N$ frames and $D$ dimensions, and $v_j$ ($j = 1, \ldots, N$) represents a frame in $v_{c,k}$. For each video clip $v_{c,k}$:

1. the manifold representation is defined in the lower-dimensional space $\mathcal{M}$ as $Vy_c = \{vy_1, vy_2, \ldots, vy_N\} \in \mathbb{R}^{N \times d}$, where $d \ll D$; and

2. the local models are then constructed as $Vp_c = \{vp_1, vp_2, \ldots, vp_M\}$, where $M \ll N$.

To overcome the inter-variations in illumination, pose, camera move, scale and other factors, each topic is represented by a set of local models rather than a single global model. These models are derived by performing a clustering within each video clip in the test set, to characterise the variations, followed by a linear fitting to each local cluster.

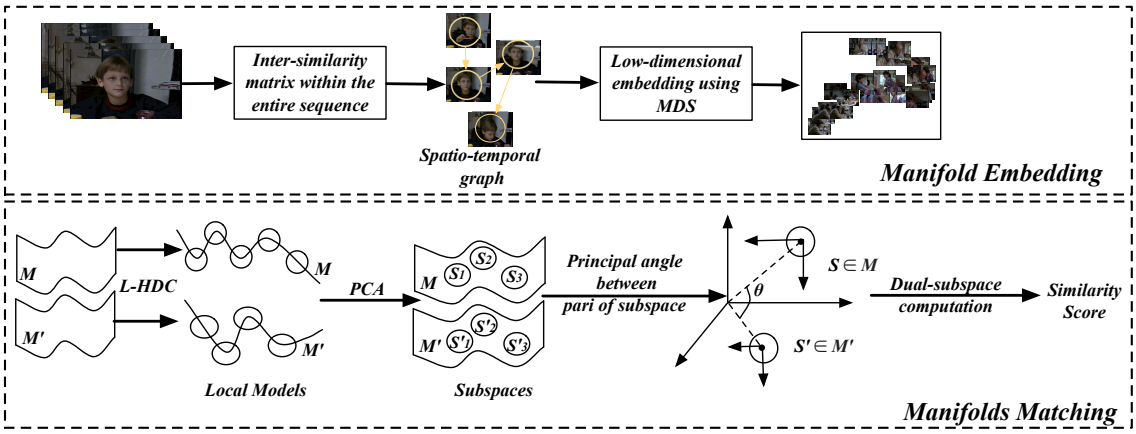



Figure 6.8: Processing steps for manifold matching using video clips.

Suppose that a query clip $X = \{x_1, \ldots, x_Q\} \in \mathbb{R}^{Q \times D}$ has one of the $C$ topics and contains $Q$ frames with $D$ dimensions, where $x_i$ represents a frame in $X$. The manifold representation for $X$ is defined using the STG-Isomap (*c.f.* Chapter 5) as: $Xy = \{xy_1, xy_2, \ldots, xy_Q\} \in \mathbb{R}^{Q \times d}$, where $d \ll D$, and the set $Xp$ with $L$ local models are computed as: $Xp = \{xp_1, xp_2, \ldots, xp_L\}$, where $L \ll Q$. A comparison is then made between each local model $xp_i$ $(i = 1, 2, \ldots, L)$ derived from the query clip $X$ and the local models $vp_j$ $(j = 1, 2, \ldots, M)$ from the $k$-th test clip $v_{c,k}$. The matching score for the $k$-th test clip is defined as:

$$score_k = \max_{1 \le j \le M} \{ \max_{1 \le i \le L} K(xp_i, vp_j) \}, \tag{6.19}$$

where $K(xp_i, vp_j)$ is the probability modelling the chance that the query clip and the test clip lie on the nearby manifolds. Finally a ranked list is created with video clips ordered by their scores.

#### 6.4.3.2 Manifold Embedding

For each video clip in the test set or the query set, the high-dimensional representation is mappede to a spatio-temporal manifold representation where nodes represent frames and edges represent the temporal order (event sequence). The method reconstructs the frames order based on their spatio-temporal relationship and recalculates distances along them to ensure the shortest distance. Given a video clip $X$ with $Q$ frames, the algorithm is presented with more detail in Chapter 5 and summarised in the following three steps:

**Step 1.** The similarity matrix $\delta$ is firstly calculated between the video frames using the Euclidean distance. The value of $\delta_{ij}$ defines the distance between two frames $x_i$ and $x_j$ $(i, j = 1, \ldots, Q)$.

**Step 2.** For each frame instance $x_i$, the $L$ frames whose distance is the closest to $x_i$ are connected. They are referred to as spatial neighbours $sn_{x_i}$. Another $L$ frames, chronologically ordered around $x_i$, are set as temporal neighbours $tn_{x_i}$. To optimise the set of temporal neighbours, $tn_{sn_{x_i}}$ is selected from temporal neighbours of spatial neighbours. The two sets of neighbours $sn_{x_i}$ and $tn_{sn_{x_i}}$ are combined, producing spatio-temporal neighbours $stn_{x_i}$ for each frame $x_i$.

**Step 3.** Given the spatio-temporal neighbourhood graph $\delta$, the distance between each pair of nodes is recalculated using the shortest path algorithm, forming a new matrix $\delta_\gamma$ of pairwise geodesic distances. Shortest paths between nodes in the graph are calculated using Dijkstra's algorithm.

**Step 4.** The multidimensional scaling [Borg and Groenen (2007)] is then applied for manifold embedding. It is formed as a transformation $T : \delta_\gamma \rightarrow Xy$ of the high-dimensional data $X$ represented by the correlation $\delta_\gamma$ into a new $d$-dimensional embedded space $Xy$.

#### 6.4.3.3 Manifolds Matching

As stated earlier, the problem of manifold matching is converted to measuring the distances between their subspaces. The first step is to model the non-linear manifolds as a collection of local linear models using the L-HDC [Wang et al. (2008)], which does not need a prior knowledge about the target clusters and consider the linearity of the space. The similarity measurement between a query video clip $X$ and a test video clip $v_{c,k}$ is defined using the principal angle [Bjoerck and Golub (1971)] and the dual-subspace method [Moghaddam et al. (2000)]. Since local models are constructed from linear patches, the canonical angles and the dual-subspace are the most suitable and efficient measurements for matching frameworks. There are different approaches for global representation, such as the mutual subspace method (MSM) [Yamaguchi et al. (1998)]. By defining the distance measurement between a pair of models, the manifold matching measurement is finally derived.

**Local linear model construction**

Various approaches have been proposed in the literature to define local models from a manifold, such as K-means [Ho et al. (2003)] and hierarchical agglomerative

clustering (HAC) [Fan and Yeung (2006)]. However these methods require the target number for clustering as prior information. In addition they do not explicitly guarantee the linearity property of the extracted models. On the other hand, L-HDC is able to provide more efficient hierarchical divisive clustering with linearity constraints to adaptively construct a multi-level model [Wang et al. (2008)].

To define the local models, MLPs are firstly extracted from the given manifold. Inspired by the geometric intuition in Tenenbaum et al. (2000), MLPs are constructed to cover as much information as possible and maximise the local neighborhood, where their linearity is reflected by the deviation between the Euclidean and the geodesic distances in the patch as shown in Figure 6.9. This is then followed by a top-down hierarchical clustering method that constructs a model as a multi-level MLP with different non-linearity degrees.

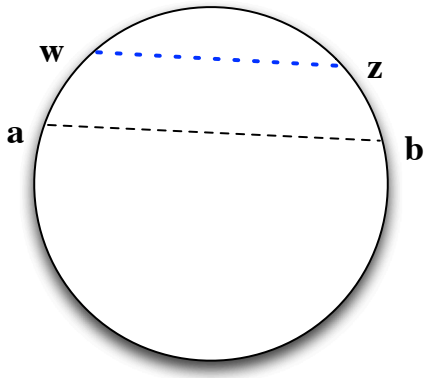
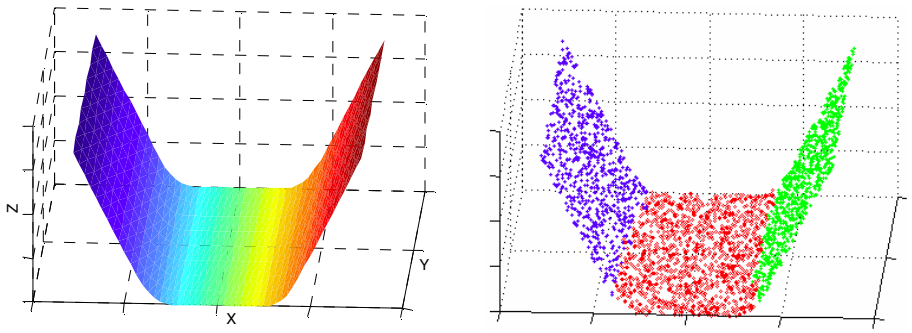



Figure 6.9: The idea of MLP linearity. Measuring the distance between $w$ and $z$ using the Euclidean distance (dashed blue line) approximates the geodesic distance (solid curve line); thus it can be discovered by MLP. However, between $a$ and $b$ the space is too curved to be defined by MLP and the Euclidean distance (dashed line) deviates too much from the geodesic distance (solid arc). Figure taken from [Wang et al. (2008)]

.

Figure 6.10 illustrates the constructed MLPs using the Wang et al. (2008) method on a "U-like shape" manifold. It is a patch-wise 3D linear manifold where points from the same plane span a linear patch, and the two adjacent planes are smoothly connected. However, manifold embedding methods would ignore this information and analyse the manifold as single space rather than as a set of patches.




**(a) 3D "U-like shape" manifold**

**(b) Three MLPs**

Figure 6.10: MLPs constructed from "U-like shape" manifold. (a) The 3D patch-wise linear manifold with "U-like shape" contains three planes connected with each other. (b) Three MLPs, represented by three different colors, is then defined later by a linear subspace. This figure is taken from [Wang et al. (2008)]

.

Formally, a data set $Xy = \{xy_1, xy_2, \ldots, xy_Q\}$ with $Q$ samples is derived from a low-dimensional manifold $\mathcal{M}$ as presented in Section 6.4.3.2. The aim is to define a set of MLPs $Xp$ (*i.e.* local models) with $L$ patches, each of which contains $Q_i$ points:

$$\begin{aligned} Xy &= \bigcup_{i=1}^{L} xp_i \\ xp_i \cap xp_j &= \phi, \quad i \neq j, \quad i, j = 1, 2, \ldots, L \\ xp_i \mid_{i=1}^{L} &= \left\{p_1^{(i)}, p_2^{(i)}, \ldots, p_{Q_i}^{(i)}\right\}, \quad \sum_{i=1}^{L} Q_i = Q. \end{aligned} \tag{6.20}$$

Then each MLP $xp_i$ is expressed as a linear subspace $S_i$ to represent the manifold $\mathcal{M}$ as a collection of subspaces:

$$\mathcal{M} = \{S_1, S_2, \ldots, S_L\}. \tag{6.21}$$

The algorithm can be summarised as follows:

1. Compute the pairwise Euclidean distance $D_E(xy_i, xy_j)$ and the geodesic distance $D_G(xy_i, xy_j)$ matrices using the kNN and the shortest path graph as in Tenenbaum et al. (2000).

2. Create the ratio-distance matrix:

$$R(xy_i, xy_j) = D_G(xy_i, xy_j) - D_E(xy_i, xy_j). \tag{6.22}$$

3. Define the neighbourhood matrix $H(:, j)$ for $j = 1, 2, \ldots, Q$ that holds the indices of the kNN points for each data point $xy_j$.

4. Initialise the first level with all the data points as a singleton MLP (cluster), *i.e.* $L = 1$ and $xp_1 = \{(p_1^{(1)})xy_1, (p_2^{(1)})xy_2, \ldots, (p_Q^{(1)})xy_Q\}$.

5. Using the ratio matrix defined as above, compute the non-linearity score $\beta_i$ for the MLP, $xp_i$ $(i = 1, 2, \ldots, L)$:

$$\beta_i = \frac{1}{Q_i^2} \sum_{t=1}^{Q_i} \sum_{z=1}^{Q_i} R\left(p_t^{(i)}, p_z^{(i)}\right). \tag{6.23}$$

6. Choose the MLP, $xp_i$ $(i = 1, 2, \ldots, L)$, with the largest score as a parent cluster. Split $xp_i$ as follows:

   (a) Based on the geodesic distance $D_G$, initialise two child clusters $p_a^{(i)}$ and $p_b^{(i)}$ with the furthest points $xy_a$ and $xy_b$ and remove them from the parent cluster $xp_i$.

   (b) Then for each child cluster, define two smaller neighbour sets $U_a$ and $U_b$ from $H$ which contain the kNN samples.

   (c) Update the parent and the child clusters by removing the points defined in neighbour sets from the parent cluster $xp_i$ and add them to the child clusters $p_a^{(i)}$ and $p_b^{(i)}$.

   (d) Split the parent cluster again into two new child sets $p_a^{(i)}$ and $p_b^{(i)}$ and start again from step (6a).

7. The entire procedure stops when the non-linearity score in step (5) is less than a predefined threshold, which controls the final number of clusters $L$ and their linearity degrees. The larger threshold gives larger linearity and fewer clusters, and vice versa. At the end, a multi-level MLPs with different non-linearity degrees is obtained.

The extracted MLPs (*i.e.* $xp_i$'s) are then represented by linear subspaces (*i.e.* $S_i$'s) to define the final local models. For each model $xp_i$, the sample mean, or the exemplar, is denoted by $e_i$ and the corresponding eigenvectors of the covariance matrix is presented by $C_i \in R^{D \times d_i}$, forming a set of orthogonal basis of the subspace with $d_i$ dimensions.

### Principal angles

Consider two subspaces $S_1$ from the query clip and $S_2$ from the test clip, with their corresponding exemplars $e_1$ and $e_2$, and their orthonormal bases $C_1 \in R^{D \times d_1}$ and $C_2 \in R^{D \times d_2}$, where $d_1$ and $d_2$ are the subspace dimensions. The principal angles $0 \leq \theta_1 \leq \ldots \leq \theta_r \leq \pi/2$ between two subspaces $S_1$ and $S_2$ are defined as the minimal angles between any two vectors of the subspaces (as shown in Figure 6.11):

$$\begin{aligned} \cos\theta_z = \max_{u_z \in S_1} \max_{v_z \in S_2} & {u_z}^T v_z \\ s.t. \quad & {u_z}^T u_z = {v_z}^T v_z = 1; \\ & {u_z}^T u_i = 0, \quad {v_z}^T v_i = 0, \quad i = 1, 2, \ldots, z-1; \\ & z = 1, 2, \ldots, r \\ & r = \min(dim(S_1), dim(S_2)), \end{aligned} \tag{6.24}$$

where $u_z$ and $v_z$ are the $z$-th pair of canonical vectors. The first constraint requires the vectors to be normalised and the second one requires the canonical vectors to be orthogonal. $\cos\theta$ calculates the canonical correlations, where the smaller the maximum value the closer the two subspaces.

Bjoerck and Golub (1971) proposed a numerically stable algorithm to compute the principal angles based on Singular Value Decomposition (SVD):

$$C_1^T C_2 = Q_1 \Lambda Q_2^T, \tag{6.25}$$

where $\Lambda = diag(\sigma_1, \ldots, \sigma_r)$, and $Q_1$ and $Q_2$ are the orthogonal matrices. The values $\sigma_1, \ldots, \sigma_r$, are the cosines of the principle angles defined by the canonical correlation:

$$\cos\theta_z = \sigma_z, \quad z = 1, 2, \ldots, r. \tag{6.26}$$

The associated canonical vectors are

$$U = C_1 Q_1 = [u_1, \ldots, u_{d_1}] \tag{6.27}$$

$$V = C_2 Q_2 = [v_1, \ldots, v_{d_2}], \tag{6.28}$$

which are defined by aligning $C_1$ and $C_2$ through an orthogonal transformation.

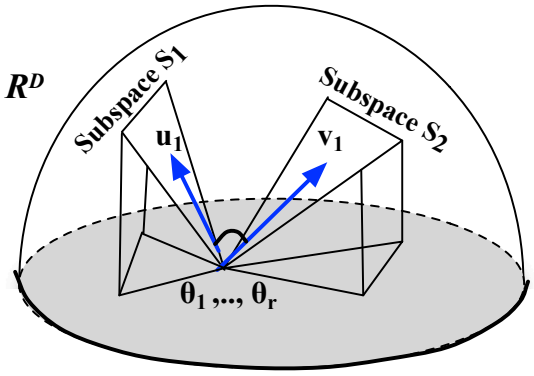



Figure 6.11: The principal angles between two subspaces $S_1$ and $S_2$ are defined as the minimum angles between any two vectors $u_1$ and $v_1$ of the subspaces.

### Dual-subspace

Following the dual-space method proposed by Moghaddam et al. (2000), consider the feature space of vectors $\triangle = v_{c,j} - v_{c,t}$, representing the differences between two videos $v_{c,j}$ and $v_{c,t}$. For each subspace $S_j$, two mutually exclusive representations are considered: intra-variation $\Omega_I$ between multiple videos of the same topic and extra-variation $\Omega_E$ for matching two different topics:

$$\Omega_I(j) = \{\triangle \mid \triangle = vp_j - \mu_j,\ \forall vp_j \in S_j\} \tag{6.29}$$

$$\Omega_E(j) = \{\triangle \mid \triangle = vp_t - \mu_j,\ \forall vp_t \in S_t,\ t \neq j\}, \tag{6.30}$$

where $\mu_j = \frac{1}{M}\sum_{q=1}^{M} vp_q$ is the centre of the subspace $S_j$ with $M$ models.

To estimate the similarity between two models, one from a query clip and another from a test clip, the difference $\triangle_i = xp_i - \mu_j$ is derived, which is then used in the

calculation of the probability $K$ in Equation (6.19):

$$K(xp_i, vp_j) = \frac{|\cos(\theta(\triangle_i, \Omega_I(j))) - \cos(\theta(\triangle_i, \Omega_E(j)))|}{|\cos(\theta(\Omega_I(j), \Omega_E(j)))|}, \tag{6.31}$$

where $\theta(\triangle_i, \Omega_I(j))$ or $\theta(\triangle_i, \Omega_E(j))$ are the largest canonical angle between $\triangle_i$ and $\Omega_I(j)$ or $\Omega_E(j)$, respectively.

### 6.4.4 Experiments

The approach was evaluated by the modified version of the NIST TRECVID instance search task. Queries and test data collection were both video clips, modelled as manifolds, and matched by seeking the maximum score as in Equation (6.19). The original TRECVID task searched for a specific entity (person, object or place) given still images, while this experiment replaced still images with video streams. Given query clips with nine (9) different topics (three from each entity), the purpose of the experiment was to retrieve four (4) similar video clip containing the same topic. The query clip contained a specific topic from one of the three entities. Video clips were identified from a small collection of 90 videos and a ranked list was created on completion of the task.

From of the set of publicly available datasets introduced in Chapter 2 (Section 2.4), the Flickr videos collection from the TRECVID 2012 task was used for this experiment [Over et al. (2008)]. It was the most suitable one since it is been used at the original INS task. In addition, to achieve the defined task, the requited videos should contain a set of recognizable entities which were easily available through this Flickr videos collection to serve the original task. It contains 74,958 short video clips with the approximate duration between 10 and 40 seconds each. Three entities (place, person and object) were provided by NIST, from which three specific topics for location ('*Eiffel Tower*', '*White House*' and '*Stonehenge*'), three specific topics for persons ('*singer*', '*broadcaster*' and '*actor*') and three specific topics for objects ('*bridge*', '*car*' and '*London tube*') were chosen. From the Flickr collection, six video clips for each of nine topics were selected, of which two were used as a query and the rest were kept for evaluation. Sample screen shots from each entity/topic were presented in Chapter 2 (Figure 2.8). Further, 12 additional video clips randomly for each entity were picked (*i.e.* the total of 36 clips), making 18 short videos in the query set and 72 short videos in the test dataset.

The ground truth was constructed for each one of the three topics (in each of the three entities) using three human judges. A subset of the dataset was browsed manually and the required topics were identified by checking the video content. This resulted in nine lists of video clips identified by the nine topics, with each list containing six video clips. In addition, 36 unrelated video clips with unknown topics were retained, to increase the dataset and add some false retrievals.

#### 6.4.4.1 Procedure and Parameter Setting

At this application there were two main steps; the manifold representation and the manifold matching that contain clustering or local models generating. To assess the performance of both steps, the approach (Framework 1) was compared with three simplified alternatives (Frameworks 2, 3 and 4):

**Framework 1** – STG-Isomap/manifold matching:
video representation was defined using the spatio-temporal graph as part of the STG-Isomap, followed by manifold matching which involved linear models construction, principal angles and dual-subspace score computation.

**Framework 2** – STG-Isomap/synchronisation map:
Manifold matching of Framework 1 was replaced with synchronisation map. The aim of this Framework is to measure the effect of the manifold matching step. The synchronisation map was chosen because it is the most suitable one for the manifold representation and for matching video contents as been used in Chapter 5.

**Framework 3** – PCA/$k$-means clustering:
This framework was defined to evaluate the entire framework. STG-Isomap of Framework 1 was replaced with the PCA as one of the most usable method in video-processing applications, and manifold matching with $k$-means clustering which is the most popular technique for content clustering.

**Framework 4** – Image intensity/manifold matching [Wang et al. (2008)]:
Image intensity was used as representation to evaluate the effect of the manifold representation step, followed by linear models construction, then computing the matching score as the average of variation distance and exemplar distance. Image intensity was chosen as the most straightforward method for video representation and would contain as much information as possible.

| Query topic | Entity | Fw 1 (this work) | Fw 2 | Fw 3 | Fw 4 |
|---|---|---|---|---|---|
| Eiffel Tower | Place | 100 | 62.5 | 25.0 | 75.0 |
| White House | Place | 100 | 87.5 | 62.5 | 100 |
| Stonehenge | Place | 75.0 | 50.0 | 37.5 | 50.0 |
| singer | Person | 75.0 | 37.5 | 37.5 | 62.5 |
| broadcaster | Person | 100 | 62.5 | 50.0 | 75.0 |
| actor | Person | 87.5 | 50.0 | 37.5 | 62.5 |
| bridge | Object | 87.5 | 62.5 | 25.0 | 50.0 |
| car | Object | 87.5 | 37.5 | 25.0 | 62.5 |
| London tube | Object | 100 | 50.0 | 37.5 | 100 |
| average (%) | | 90.3 | 55.6 | 37.5 | 70.8 |

Table 6.5: Instance search experiment: comparison of four frameworks, denoted by Fw 1, 2, 3 and 4, were made using the Flickr video dataset. Each query was given by a video clip, presenting a topic that belonged to one of three entities (place, person or object). There were nine topics and, for each topic, four relevant clips should have been identified from the collection of 72 videos. The average was the score average calculated by the matching result for individual topics over the queries number.

Video representation was created as follows. For the local features extraction, the ST-SIFT detector developed in Chapter 3 was firstly applied to detect the interest points containing informative spatial locations as well as interesting temporal information. Spatio-temporal regions around the interest points were described using HOG [Scovanner et al. (2007)]. The extended LLC coding presented in Chapter 4 was then utilised to encode spatio-temporal features into more compact and independent components. For the spatio-temporal graph construction, the initial number of neighbours frames appeared dependent on the clip length and was selected manually.

#### 6.4.4.2 Results and Analysis

Table 6.5 presents the results from the video query based instance search experiment, comparing Framework 1 against Frameworks 2, 3 and 4. The final score for each query clip was computed based on Equation (6.19) and then the final score for each topic was computed as the summation of query clips scores divided by the number of queries ($Q_x$):

$$score_{topic} = \frac{1}{Qx}\sum_{i=1}^{Qx}\left[\sum_{k=1}^{F} score_k\right]. \tag{6.32}$$

The results show that the approach presented in this paper (Framework 1 – STG-Isomap/manifold matching) achieved over 90%, outperforming its simplified alternatives by a fair margin (34.7, 52.8 and 19.5% absolute against Frameworks 2, 3 and 4). The main reasons were:

- Defining the intra-correlation within the video sequence, using the graph-based representation STG-Isomap, helped to reconstruct the video frames in the manifold in order to uncover similarities within the video. Clips from the same entity were mapped close to each other, resulting in similar representations in the manifold.

- The manifold matching step treated both query and test data as manifolds, and this helped to measure the similarity not only within the data itself but also between their variations. Combining the MLP and L-HDC clustering methods solved a number of potential problems, such as unbalanced clustering and the extent of linearity.

- Unlike the other methods such as $k$-means clustering, which are sensitive to the initial parameters and might have failed in the local minima, the L-HDC algorithm adapted in the approach was more stable with the variation of data.

Framework 2 (STG-Isomap/synchronisation map) outperformed Framework 3 with the object entity (query topics 1 to 3 in Table 6.5). These scenes contained camera or objects moves, thus causing discontinuity within the sequence because they did not share sufficient numbers of spatial features with their neighbours. Consideration of the temporal relation alleviated this problem.

Framework 3 (PCA/$k$-means clustering) was based on an application of PCA followed by the $k$-means clustering method. Euclidean distances were computed between the cluster centres, which were treated as local models. Of all the frameworks, this combination exhibited the lowest score. This was expected since the linearity was not warranted explicitly with this local models (*i.e.* clusters). Additionally, these approaches were likely to fail with images of poor quality while good quality images might cluster as outliers.

Finally, Framework 4 (Image intensity/manifold matching [Wang et al. (2008)]) performed better than Frameworks 2 and 3 for indoor scenes containing a person entity. This was because of only little moves taking place in the video clip and, despite various camera angles being used, there were not many dramatic changes in the scene

(either in the foreground or in the background). Under such conditions Framework 4 was able to distinguish between multiple clips spatially, projecting them onto the lower-dimensional space.

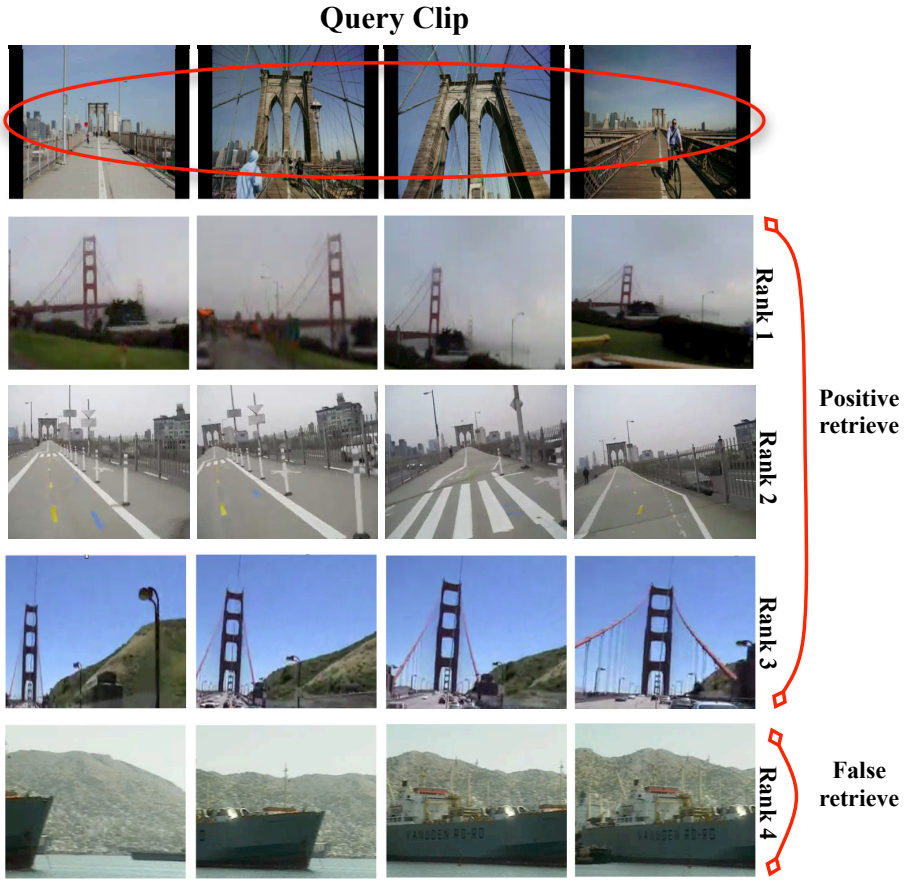



Figure 6.12: The query (top row) was one of four clips identified as '*bridge*'. Three relevant clips were ranked first, second and third out of 45 candidates. After that, false rankings were produced with the fourth one, which is a sea view from the set of the random clips added to increase the dataset.

Figures 6.12 and 6.13 show two ranking examples produced by Framework 1; for each figure the first row was the query clip and the three highest ranking clips are presented from the second row. In the pool of candidates there were scenes containing entities with different topics; however, these occurred in similar locations. Similarity calculation often clustered these scenes close to each other in the manifold. Another interesting example was the semantic similarities between different scenes with conflicting topics. This example is illustrated in Figure 6.13 with a query of the '*car*' topic where the secondly ranked clip also contained a moving object but from the different

topic. Although Framework 1 created good ranked lists, such scenario often caused erroneous ranking of unrelated entities and topics.

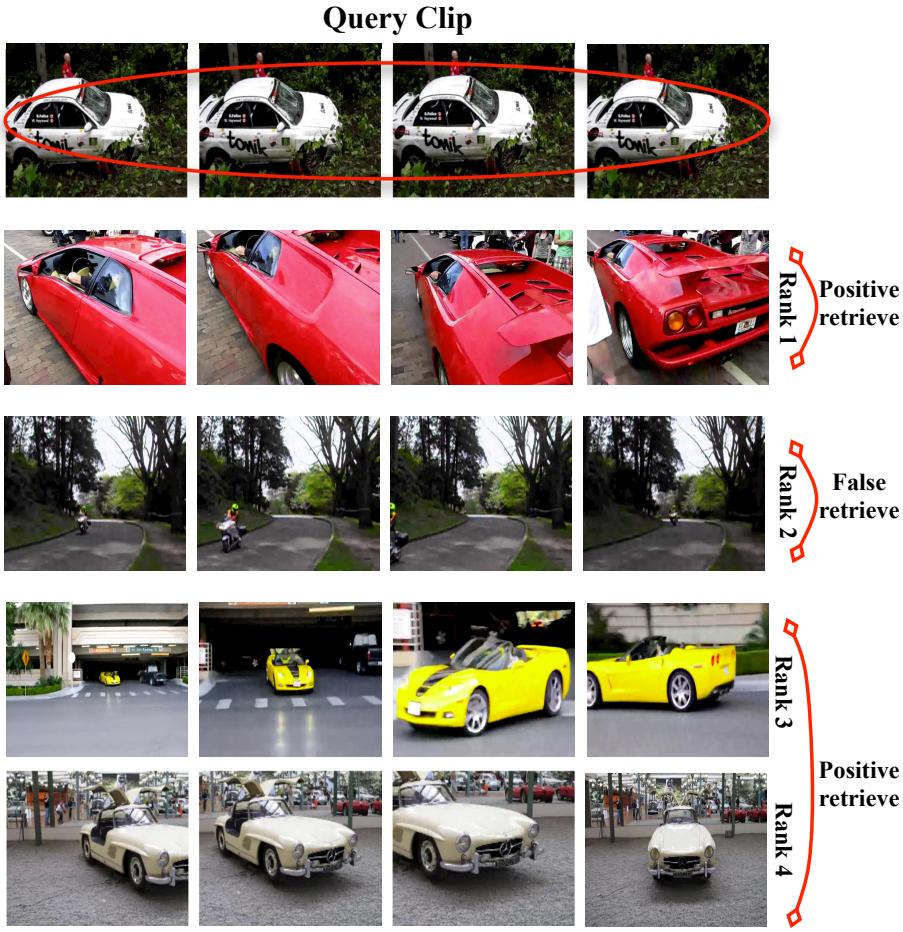



Figure 6.13: The query (top row) was one of four clips identified as '*car*'. The first, third and the fourth ranks were all relevant clips (true positives). The second rank was the false positive caused by the semantic similarity between the query and this clip containing the different type of moving objects.

## 6.5 Summary

The previous three chapters presented a video sequence alignment framework with three stages including ST-SIFT as a spatio-temporal interest points detector, ST-LLC as features learning technique and STG-Isomap as a spatio-temporal graph-based manifold embedding method. During the development, the first two stages were evaluated using the human action classification task, while the third stage was evaluated using

the nearly-repetitive alignment task. In this chapter, we evaluated the complete video sequence alignment framework by developing three applications related to the video similarity task using different datasets.

The first application was a video clip retrieval task using the EMD. The video representation was firstly defined using the ST-SIFT detector combined with ST-LLC coding. A graph-based representation was then constructed using the spatio-temporal graph construction phase at the STG-Isomap. The video clips' similarity was then measured using the distance between a pair of graphs. The EMD measured the video clip similarity by computing the minimum transportation cost within the graph. Experimental results using the challenging UCF11 dataset indicated that the approach was more capable of retrieving the relevant video clips than existing techniques.

The second application was to detect nearly-repetitive contents in a video stream using manifold embedding. The similarities observed in frame sequences were captured by defining two types of correlation graphs. The first was the intra-correlation graph within a single sequence using the ST-SIFT detector combined with ST-LLC coding techniques to densely extract and encode salient feature points from a 3D signal. The second was the inter-correlation graph between two repetitive sequences derived as a step for the STG-Isomap manifold learning. A synchronisation map between the two clips was then calculated and used to rank the repetitive contents. Experimental results using rushes video showed that the approach with ST-LLC performed better than the conventional manifold embedding techniques.

The last application addressed the problem of matching video clips that contained recognisable entities defined by the query clip. A presented framework was utilised to align two non-linear manifolds based on their local linear models. Video representation was firstly constructed using ST-SIFT combined with ST-LLC. A spatio-temporal graph was derived as a step for the STG-Isomap manifold embedding that defined the intra-correlation across video sequences. The local linear models were then extracted using the hierarchical clustering method. The principal angles between local models, belonging to a pair of manifolds, and the similarity score were then calculated. The manifolds matching was finally derived by defining the distance measurement between a pair of models. Experimental results using video clips collected from Flickr showed that the presented approach with spatio-temporal representation performed better than conventional techniques.

# Chapter 7

# Conclusions

Video sequences alignment is a demand for various computer vision applications that process multiple videos simultaneously. The alignment task seeks to capture the relationship between video content in the spatio-temporal domain. It is a fundamental task that helps with spatial ambiguities and covers situations that cannot be covered by image-processing approaches. Defining a video representation is the core step for video-processing applications including sequences alignment. In video data a frame sequence is chronologically tied to present a story; each frame captures a moment related to the adjacent frames. Their spatial and temporal relationships are embedded in the high-dimensional space, generating a complex structure of video. The video content was analysed in this thesis by characterising the spatial and temporal information embedded in the high-dimensional space through the frame sequence. The problem of measuring the similarity between a pair of video sequences is formulated as finding the correspondences between their feature trajectories in the lower-dimensional space.

This thesis is concerned with the video sequences representation and alignment framework that can be used for video indexing, retrieval and summarisation applications. Initially, a spatio-temporal interest points detector was developed to reduce the video from a volume of pixels to a descriptive features trajectory (Chapter 3). These points have significant local variations in both spatial and temporal domains and are invariant to different spatio-temporal variations such as scale, location and orientation. Secondly, the generated descriptors were quantised to define the intra-correlation between frames within each video sequence (Chapter 4). This correlation captures the relationship between the frames of each video, reorders and clusters the video content into groups of frames in the context of spatio-temporal similarity. Finally a manifold embedding technique was developed to synchronise and align video content in

lower-dimensional space (Chapter 5). At this point, the inter-correlation is defined between multiple video sequences using the spatio-temporal neighbours graph. This correlation captures the relationship between a pair of video sequences based on their spatio-temporal similarity. Many problems can be solved by analysing the defined representations in the lower-dimensional space, which contain an ordered frames by the spatio-temporal similarity and connected as trajectories to be used for sequences alignment.

For the completed video sequences alignment framework, three applications related to the video similarity, searching and retrieval tasks were developed to verify the approach's ability in various real-life video applications. (Chapter 6). Further, multiple types of datasets with different characteristics were employed in the development and evaluations to measure the framework's ability in dealing with variations in scales, viewing angles, background conditions (indoors and outdoors), etc.

The first application was a video clip retrieval task that adopted the many-to-many graph-matching method from the image-processing domain to measure the similarity between two spatio-temporal graphs in the lower-dimensional space. The second was a video clip ranking task to identify nearly-repetitive contents in the video sequences, where the original material is transformed to nearly, but not exactly, identical contents. For that a synchronisation map is calculated for the video contents to discover the underlying structure in the manifold. The last application was an instance search based on video queries that aim to locate clips with a recognisable entity in a collection of test video clips. For that, each manifold was represented as a set of locally linear models, each of which was interpreted as a subspace. The manifold matching is then solved by measuring the similarity between a pair of subspaces, one from each of the manifolds.

The motivation of this research comes from two sides. First, every day a wide range of multimedia have been recorded without processing, which gives rise to potential problems for content management to serve user information needs. To solve the accessing problem, many algorithms have been proposed to represent, search and retrieve visual content in multimedia data. However, there is still a big challenge to access video content efficiently and to help users gain useful information relevant to their interests. Second, observing the data over time provides a unique view for analyses. Most activities and video events can be detected through the frame sequence rather than from individual frames. Scene dynamics (e.g. object movement), non-rigid changes in the scene (e.g. flowing water) and changes in illumination are examples of

information found between the frames and that cannot be captured by representing individual frames. All these observations inspired this thesis: to study the video representation task from both space and time domains, and the sequences alignment task at the lower dimensional space.

## 7.1 Original Contributions

1. **Video sequence alignment framework:** A three-stage framework was developed to align video sequences in a lower-dimensional space using a spatio-temporal interest points detector, coding technique and manifold embedding. Most previous works related to sequence-to-sequences alignment utilised template matching, or required prior information such as video length and contents. Here, the video content was analysed by capturing the spatial and temporal correlation embedded in a frame sequence. A video was represented with spatio-temporal features, encoded with more representative and compact linear codes and finally projected into the lower-dimensional space to reconstruct the underlying structure so that similar contents can be reorganised in the manifold. Evaluation in Chapter 6 with video searching, retrieval and ranking tasks using different datasets showed that the presented framework achieved a significant improvement over the conventional alternatives in capturing the similarity between videos .

2. **Space-time video clip representation:** The first stage was presented in Chapter 3, which delivered the ST-SIFT detector to extract interest points that have significant local variations in both space and time domains and are invariant to scale, location and orientation. Spatio-temporal Gaussian and DoG pyramids were firstly constructed. The generated DoG volume was then segmented into three spatial and temporal planes ($xy$, $xt$ and $yt$) and the common points between these planes were chosen as interest points. The derived representation was evaluated in the action recognition task, which is the core task for feature extraction techniques, and achieved results comparable to those of state-of-the-art methods.

3. **Coding technique:** The second stage, described in Chapter 4, was to quantise the generated descriptors using ST-LLC that defines intra-correlation between frames within each video sequence. This correlation defines the relationship

between the frames of each video that reorder and cluster the video content into a group of frames in the context of spatio-temporal similarity. It was derived by firstly constructing a spatio-temporal graph between the descriptors and the codebook, then calculating the shortest-path and performing a kNN search and finally solving a constrained least-squares fitting problem. The coding stage represents video content with a mid-level representation containing fewer codes than the original set of low-level features, which helps to save both processing time and storage space for the visual descriptors.

4. **Manifold Representation:** The final stage of the alignment framework was the development of STG-Isomap manifold embedding in Chapter 5 to synchronise and align video sequences in lower-dimensional space. At this stage the inter-correlation was computed between multiple video sequences using the spatio-temporal neighbours graph. This correlation defines the relationship between a pair of video sequences by reordering and clustering their frames into groups based on spatio-temporal similarity. It involved defining three sets of neighbours including spatial based on the distance, temporal based on the frames order, and temporal of the spatial. The union between them is defined as spatio-temporal neighbours, the shortest path is computed and finally the dimensionality reduction is applied. The generated coordinates in the lower-dimensional space were ordered according to the similarity and integrated to define a features trajectory representation for sequence alignment.

## 7.2 Future Work

There are many future research opportunities in video sequences alignment and the more general research area of video similarity. We categorise future research into four directions: spatio-temporal video representation, features coding, manifold embedding, and video content understanding.

1. ST-SIFT was extended from SIFT, aiming to transfer its robustness from image-processing to video-processing, and it gave promising results as spatio-temporal features to describe video content. However, ST-SIFT also inherits the shortcoming of SIFT. Interest points detected by ST-SIFT are the high contrast points at different scales, which sometimes is not enough to describe some activities. In addition, ST-SIFT extracted the features from gray-scale frames, resulting in

lots of colour information being discarded. Further, dealing with large datasets can be time-consuming, because of the consequent large amount of computation of ST-SIFT and its variants. All these weaknesses need further research, to enhance or extend ST-SIFT to be a more robust video feature.

2. There are several ways the efficiency of the coding stage may be improved. The kNN data structure can be replaced with another method such as kd-tree, ball-tree, or cover-tree to speed up the search. Learning the codebook using other techniques, such as supervised dictionary learning, can also be considered to enhance the performance. In addition, the inner structure and relationship of the codebook entries can be investigated to explore the semantic correlation between them.

3. STG-Isomap in its current development seeks to uncover spatio-temporal video content by processing all its frames. For large datasets, we may either process a subset of the data or apply an interval segmentation of the data. It may also be worth investigating uncovering the spatio-temporal structure without necessarily calculating shortest-paths between every frames-pair or embedding a full distance matrix. For the present work, the neighborhood function, size and the number of embedding dimensions to retain were empirically selected. The performance may improve if the selection can be adaptively defined.

4. The proposed alignment framework has a solid performance in various tasks. On the other hand, it can still be improved. The most obvious topic is to segment video contents into semantic scenes that may limit the search space and enhance the accuracy of similarity detection. Defining semantically coherent scenes can be a core step in understanding video content. Additionally, dividing the video sequence into semantic contents helps in the indexing and browsing of video data as well as in processing large datasets with long video sequences.

# References


Aggarwal, J. and Ryoo, M. (2011). Human activity analysis: A review. *ACM Computing Surveys*, pages 70–80.

Agusti, P., Traver, V. J., and Pla, F. (2014). Bag-of-words with aggregated temporal pair-wise word co-occurrence for human action recognition. *Pattern Recognition Letters*, 49:224 – 230.

Al Ghamdi, M., Al Harbi, N., and Gotoh, Y. (2012a). Spatio-temporal video representation with locality-constrained linear coding. In *Proceedings of the European Conference on Computer Vision. Workshops and Demonstrations*, ECCV, pages 101–110.

Al Ghamdi, M. and Gotoh, Y. (2013). Spatio-temporal manifold embedding for nearly-repetitive contents in a video stream. In *Proceedings of the 15th International Conference on Computer Analysis of Images and Patterns*, CAIP.

Al Ghamdi, M. and Gotoh, Y. (2014a). Alignment of nearly-duplicate contents in video stream with manifold embedding. In *Proceedings of the 39th International Conference on Acoustics, Speech and Signal Processing*, ICASSP, pages 1255–1259.

Al Ghamdi, M. and Gotoh, Y. (2014b). Manifold matching with application to instance search based on video queries. In *Proceedings of the 6th International Conference on Image and Signal Processing*, ICISP, pages 477–486.

Al Ghamdi, M. and Gotoh, Y. (2014c). Video clip retrieval by graph matching. In *Proceedings of the 36th European Conference on Information Retrieval*, ECIR, pages 412–417.

Al Ghamdi, M., Zhang, L., and Gotoh, Y. (2012b). Spatio-temporal SIFT and its application to human action classification. In *Proceedings of the European Conference on Computer Vision. Workshops and Demonstrations*, ECCV, pages 301–310.

Allaire, S., Kim, J., Breen, S., Jaffray, D., and Pekar, V. (2008). Full orientation invariance and improved feature selectivity of 3D SIFT with application to medical image analysis. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, CVPR.

Arfken, G. B. and Weber, H. J. (2005). *Gram-Schmidt Orthogonalization*. Academic Press.

## References


Bailer, W., Lee, F., and Thallinger, G. (2007). Skimming rushes video using retake detection. In *Proceedings of the international workshop on* TRECVID *video summarization*, TVS, pages 60–64.

Bay, H., Tuytelaars, T., and Gool, L. (2006). SURF: Speeded up robust features. In *Proceedings of European Conference on Computer Vision*, ECCV, pages 404–417.

Belkin, M. and Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, pages 1373–1396.

Benini, S., Canini, L., Migliorati, P., and Leonardi, R. (2009). Multimodal space for rushes representation and retrieval. In *Proceedings of the 7th International Workshop Content-Based Multimedia Indexing.*, CBMI, pages 50 –55.

Bergen, J., Burt, P., Hingorani, R., and Peleg, S. (1992). A three-frame algorithm for estimating two-component image motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 886–896.

Bjoerck, A. and Golub, G. H. (1971). Numerical methods for computing angles between linear subspaces. Technical report, Stanford University, Stanford, CA, USA.

Blank, M., Gorelick, L., Shechtman, E., Irani, M., and Basri, R. (2005). Actions as space-time shapes. In *Proceedings of the IEEE International Conference on Computer Vision*, ICCV, pages 1395–1402.

Borg, I. and Groenen, P. (2007). *Modern Multidimensional Scaling: Theory and Applications.* Springer Series in Statistics. Springer.

Boureau, Y.-L., Bach, F., LeCun, Y., and Ponce, J. (2010). Learning mid-level features for recognition. In *Proceedings of the International Conference Pattern Recognition*, CVPR, pages 2559 –2566.

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization.* Berichte über verteilte messysteme. Cambridge University Press.

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

Campos, T., Barnard, M., Mikolajczyk, K., Kittler, J., Yan, F., Christmas, W. J., and Windridge, D. (2011). An evaluation of bags-of-words and spatio-temporal shapes for action recognition. In *Proceedings of the IEEE Workshop on Applications of Computer Vision*, WACV, pages 344–351.

Carroll, J. and Chang, J.-J. (1970). Analysis of individual differences in multidimensional scaling via an n-way generalization of "eckart-young" decomposition. *Psychometrika*, pages 283–319.

Caspi, Y. and Irani, M. (2002). Spatio-temporal alignment of sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:1409–1424.

Chantamunee, S. and Gotoh, Y. (2010). Nearly-repetitive video synchronisation using nonlinear manifold embedding. In *Proceedings of the 39th International Conference on Acoustics, Speech and Signal Processing*, ICASSP.

Chen, L. and Chua, T. S. (2001). A match and tiling approach to content-based video retrieval. In *Proceedings of IEEE International Conference on Multimedia and Expo*, ICME, pages 301–304.

Chen, M.-y. and Hauptmann, A. (2009). MoSIFT : Recognizing human actions in surveillance videos. *Transform*, pages 1–16.

Cheung, S. S. and Zakhor, A. (2003). Efficient video similarity measurement with video signature. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 59–74.

Cheung, W. and Hamarneh, G. (2007). N-SIFT: N-dimensional scale invariant feature transform for matching medical images. In *Proceedings of IEEE International Symposium Biomedical Imaging: From Nano to Macro*, pages 720–723.

Chiu, C. Y. Yang, C. C. and Chen, C. S. (2007). Efficient and effective video copy detection based on spatiotemporal analysis. In *Proceedings of the IEEE International Symposium on Multimedia*, ISM, pages 202 –209.

Cour, T., Srinivasan, P., and Shi, J. (2007). Balanced graph matching. In *Advances in Neural Information Processing Systems 19*, pages 313–320. MIT Press.

Cox, T. and Cox, A. (2000). *Multidimensional Scaling, Second Edition*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Taylor & Francis.

Csurka, G., Dance, C. R., Fan, L., Willamowski, J., and Bray, C. (2004). Visual categorization with bags of keypoints. In *Proceedings of the 4th European Conference on Computer Vision*, ECCV, pages 1–22.

Doctor, M., Moreno, A., Muñoz, P., Díaz, D., and R-Moreno, M. D. (2011). Intelligent social networks. In *Proceedings of the International Conference on Web Intelligence, Mining and Semantics*, WIMS, pages 69:1–69:8.

Dollar, P., Rabaud, V., Cottrell, G., and Belongie, S. (2005). Behavior recognition via sparse spatio-temporal features. In *Proceedings of 2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 65–72.

Donoho, D. L. (2000). High-dimensional data analysis: The curses and blessings of dimensionality. Technical report, Stanford University.

Donoho, D. L. and Grimes, C. (2003). Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *Proceedings of the National Academy of Sciences*, pages 5591–5596.

Dorr, M., Jarodzka, H., and Barth, E. (2010). Space-variant spatio-temporal filtering of video for gaze visualization and perceptual learning. In *Proceedings of the Symposium on Eye-Tracking Research and Applications*, ETRA, pages 307–314.

Duan, L.-Y., Xu, M., Chua, T.-S., Tian, Q., and Xu, C.-S. (2003). A mid-level representation framework for semantic sports video analysis. In *Proceedings of the 11th ACM International Conference on Multimedia*, MULTIMEDIA, pages 33–44.

Dumont, E. and Merialdo, B. (2009). Rushes video parsing using video sequence alignment. In *Proceedings of the 7th International Workshop Content-Based Multimedia Indexing.*, CBMI, pages 44 –49.

Elkan, C. (2003). Using the Triangle Inequality to Accelerate k-Means. In *International Conference on Machine Learning*, pages 147–153.

Escobar, M.-J., Masson, G., Vieville, T., and Kornprobst, P. (2009). Action recognition using a bio-inspired feedforward spiking network. *International Journal of Computer Vision*, pages 284–301.

Fan, W. and Yeung, D. Y. (2006). Locally linear models on face appearance manifolds with application to dual-subspace based classification. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, CVPR.

Feng, D., Siu, W., and Zhang, H. (2003). *Multimedia Information Retrieval and Management: Technological Fundamentals and Applications*. Engineering online library. Springer.

Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, pages 381–395.

Fitzgibbon, A. W. and Zisserman, A. (2003). Joint manifold distance: A new approach to appearance based clustering. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, CVPR.

Gibbons, A. (1985). *Algorithmic Graph Theory*. Cambridge University Press.

Giese, M. and Poggio, T. (1999). Synthesis and recognition of biological motion patterns based on linear superposition of prototypical motion sequences. In *Proceedings of IEEE Workshop on Multi-View Modeling and Analysis of Visual Scenes*, MVIEW, pages 73–80.

Gilbert, A., Illingworth, J., and Bowden, R. (2009). Fast realistic multi-action recognition using mined dense spatio-temporal features. In *Proceedings of IEEE International Conference on Computer Vision*, ICCV, pages 925–931.

Gray, R. (1984). Vector quantization. *IEEE ASSP Magazine*, pages 4–29.

Hampel, F., Ronchetti, E., Rousseeuw, P., and Stahel, W. (2011). *Robust Statistics: The Approach Based on Influence Functions*. Wiley.

Harris, C. and Stephens, M. (1988). A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, pages 147–151.

Hauptmann, A., Yan, R., Lin, W.-H., Christel, M., and Wactlar, H. (2007). Can high-level concepts fill the semantic gap in video retrieval? a case study with broadcast news. *IEEE Transactions on Multimedia*, pages 958–966.

He, X. and Niyogi, P. (2003). Locality preserving projections. In *Advances in Neural Information Processing Systems*.

Ho, J., Yang, M. H., and Kriegman, D. (2003). Video-based face recognition using probabilistic appearance manifolds. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, CVPR.

Hu, W., Xie, N., Li, L., Zeng, X., and Maybank, S. (2011). A survey on visual content-based video indexing and retrieval. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, pages 797–819.

Jain, M., Jegou, H., and Bouthemy, P. (2013). Better exploiting motion for better action recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, CVPR, pages 2555–2562.

Jenkins, O. C. and Matarić, M. J. (2004). A spatio-temporal extension to isomap nonlinear dimension reduction. In *Proceedings of the International Conference on Machine Learning*, ICML, pages 441–448.

Jhuang, H., Serre, T., Wolf, L., and Poggio, T. (2007). A biologically inspired system for action recognition. In *Proceedings of IEEE International Conference on Computer Vision*, ICCV, pages 1–8.

Jiang, W., Chan, K. L., Li, M., and Zhang, H. (2005). Mapping low-level features to high-level semantic concepts in region-based image retrieval. In *Proceedings of the International Conference Pattern Recognition*, CVPR, pages 244–249.

Joly, A., Buisson, O., and Frelicot, C. (2007). Content-based copy retrieval using distortion-based probabilistic similarity search. *IEEE Transactions on Multimedia*.

Kadir, T. and Brady, M. (2001). Saliency, scale and image description. *International Journal of Computer Vision*, pages 83–105.

Ke, Y., Sukthankar, R., and Hebert, M. (2005). Efficient visual event detection using volumetric features. In *Proceedings of International Conference on Computer Vision*, ICCV, pages 166 – 173.

Kläser, A., Marszałek, M., Laptev, I., and Schmid, C. (2010). Will person detection help bag-of-features action recognition? Technical report, INRIA Grenoble - Rhône-Alpes, 655, avenue de l'Europe, 38334 Montbonnot Saint Ismier, FRANCE.

Klaser, A., Marszałek, M., and Schmid, C. (2008). A spatio-temporal descriptor based on 3D-gradients. In *Proceedings of British Machine Vision Association*, BMVC, pages 995—-1004.

Koniusz, P., Yan, F., and Mikolajczyk, K. (2013). Comparison of mid-level feature coding approaches and pooling strategies in visual concept detection. *Computer Vision and Image Understanding*, pages 479 – 492.

Kovashka, A. and Grauman, K. (2010). Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In *Proceedings of IEEE Conference Computer Vision and Pattern Recognition*, CVPR, pages 2046–2053.

Laptev, I. (2005). On space-time interest points. *International Journal of Computer Vision*, pages 107–123.

Laptev, I. and Lindeberg, T. (2003). Space-time interest points. In *Proceedings of the IEEE International Conference on Computer Vision*, ICCV, pages 432–439.

Laptev, I., Marszalek, M., Schmid, C., and Rozenfeld, B. (2008). Learning realistic human actions from movies. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, CVPR, pages 1–8.

Law, H. C. (2006). *Clustering, Dimensionality Reduction, and Side Information*. PhD thesis, Michigan State University.

Lazebnik, S., Schmid, C., and Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of the International Conference Pattern Recognition*, CVPR, pages 2169–2178.

Le, Q. V., Zou, W. Y., Yeung, S. Y., and Ng, A. Y. (2011). Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, CVPR, pages 3361–3368.

Lee, H., Battle, A., Raina, R., and Ng, A. Y. (2007). Efficient sparse coding algorithms. In *Proceedings of the Annual Conference on Neural Information Processing Systems*, NIPS, pages 801 –808.

Lee, J. and Verleysen, M. (2007). *Nonlinear Dimensionality Reduction*. Springer.

Lie, W.-N. and Hsiao, W.-C. (2002). Content-based video retrieval based on object motion trajectory. In *Proceedings of IEEE Workshop on Multimedia Signal Processing*, pages 237–240.

Lin, T. and Zha, H. (2008). Riemannian manifold learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 796–809.

Liu, J., Luo, J., and Shah, M. (2009a). Action recognition in unconstrained amateur videos. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, ICASSP, pages 3549–3552.

Liu, J., Luo, J., and Shah, M. (2009b). Recognizing realistic actions from videos in the wild. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, CVPR, pages 1996–2003.

Liu, J. and Shah, M. (2008). Learning human actions via information maximization. In *Proceedings of the International Conference Pattern Recognition*, CVPR.

Liu, L., Wang, L., and Liu, X. (2011). In defense of soft-assignment coding. In *Proceedings of the International Conference on Computer Vision*, ICCV, pages 2486–2493.

Liu, Y., Liu, Y., and Chan, K. (2008). Multiple video trajectories representation using double-layer isometric feature mapping. In *Proceedings of IEEE International Conference on Multimedia and Expo*, pages 129 –132.

Lloyd, S. (2006). Least squares quantization in PCM. *IEEE Transactions on Information Theory*, pages 129–137.

Long, F., Zhang, H., and Feng, D. D. (2003). Fundamentals of content-based image retrieval. In *Multimedia Information Retrieval and Management*, Signals and Communication Technology, pages 1–26. Springer Berlin Heidelberg.

Lopes, A., Oliveira, R., de Almeida, J., and de A Araujo, A. (2009). Spatio-temporal frames in a bag-of-visual-features approach for human actions recognition. In *Proceedings of Computer Graphics and Image Processing, XXII Brazilian Symposium*, pages 315–321.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, pages 91–110.

Lucas, B. D. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, IJCAI, pages 674–679.

Martinez, A. M. and Kak, A. (2001). PCA versus LDA. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 228–233.

Mikolajczyk, K. and Schmid, C. (2004). Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, pages 63–86.

Moghaddam, B., Jebara, J., and Pentland, A. (2000). Bayesian face recognition. *Pattern Recognition*.

Morel, J. and G.Yu (2011). Is SIFT scale invariant? *Inverse Problems and Imaging (*IPI*)*, page 115–136.

MPEG video group (1999). Description of core experiments for MPEG-7 color/texture descriptions. Technical report, ISO/MPEGJTC1/SC29/WG11 MPEG98/M2819.

Nga, D. H. and Yanai, K. (2013). A spatio-temporal feature based on triangulation of dense surf. In *Proceedings of IEEE International Conference on Computer Vision Workshops*, ICCVW, pages 420–427.

Niebles, J., Wang, H., and Fei-Fei, L. (2008). Unsupervised learning of human action categories using spatial-temporal words. *International Journal of Computer Vision*, pages 299–318.

Ning, H., Hu, Y., and Huang, T. (2007). Searching human behaviors using spatial-temporal words. In *Proceedings of IEEE International Conference on Image Processing*, ICIP, pages 337 –340.

Noguchi, A. and Yanai, K. (2012). A surf-based spatio-temporal feature for feature-fusion-based action recognition. In *Proceedings of the 11th European Conference on Trends and Topics in Computer Vision*, ECCV, pages 153–167.

Nowozin, S., Bakir, G., and Tsuda, K. (2007). Discriminative subsequence mining for action classification. In *Proceedings of IEEE International Conference on Computer Vision*, ICCV.

Oikonomopoulos, A., Patras, I., and Pantic, M. (2005). Spatiotemporal salient points for visual recognition of human actions. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, pages 710–719.

Ojala, T., Pietikäinen, M., and Mäenpää, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 971–987.

Olshausen, B. A. and Fieldt, D. J. (1997). Sparse coding with an overcomplete basis set: a strategy employed by V1. *Vision Research*, pages 3311 –3325.

Oshin, O., Gilbert, A., Illingworth, J., and Bowden, R. (2008). Spatio-temporal feature recogntion using randomised ferns. In *Proceedings of ECCV Workshop on Machine Learning for Vision-based Motion Analysis*, MLVMA.

Over, P., Awad, G., Michel, M., Fiscus, J., Sanders, G., Shaw, B., Kraaij, W., Smeaton, A. F., and Quenot, G. (2012). TRECVID 2012 — an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *Proceedings of* TRECVID.

Over, P., Smeaton, A. F., and Awad, G. (2008). The TRECVID 2008 BBC rushes summarization evaluation. In ACM TRECVID *Video Summarization Workshop*.

Ozuysal, M., Fua, P., and Lepetit, V. (2007). Fast keypoint recognition in ten lines of code. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, CVPR, pages 1–8.

Poppe, R. (2010). A survey on vision-based human action recognition. *Image and Vision Computing*, pages 976–990.

Poppe, R. W. (2009). *Discriminative vision-based recovery and recognition of human motion*. PhD thesis, University of Twente, Enschede.

Poston, T. and Stewart, I. (1978). *Catastrophe Theory and Its Applications*. Dover Books on Mathematics. Dover Publications.

Rao, C., Gritai, A., Shah, M., and Syeda-Mahmood, T. (2003). View-invariant alignment and matching of video sequences. In *Proceedings of IEEE International Conference on Computer Vision*, pages 939–945.

Rapantzikos, K., Avrithis, Y., and Kollias, S. (2007). Spatiotemporal saliency for event detection and representation in the 3D wavelet domain: Potential in human action recognition. In *Proceedings of the 6th ACM International Conference on Image and Video Retrieval*, CIVR, pages 294–301.

Reid, I. D. and Zisserman, A. (1996). Goal-directed video metrology. In *Proceedings of the 4th European Conference on Computer Vision*, ECCV, pages 647–658.

Ren, R., Punitha, P., and Jose, J. (2008). Video redundancy detection in rushes collection. In *Proceedings of the 2nd ACM* TRECVid *Video Summarization Workshop*, TVS, pages 65–69.

Rodriguez, M., Ahmed, J., and Shah, M. (2008). Action MACH a spatio-temporal maximum average correlation height filter for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, CVPR.

Roweis, S. (1998). Em algorithms for PCA and SPCA. In *Proceedings of the Annual Conference on Neural Information Processing Systems*, NIPS, pages 626–632.

Roweis, S. T. and Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *SCIENCE*, pages 2323–2326.

Roweis, S. T. and Saul, L. K. (2001). An introduction to locally linear embedding. Technical report, AT &T Labs and Gatsby Computational Neuroscience Unit, UCL.

Rubner, Y., Tomasi, C., and Guibas, L. J. (2000). The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision.*

Schindler, K. and van Gool, L. (2008). Action snippets: How many frames does human action recognition require? In *Proceedings of IEEE Conference Computer Vision and Pattern Recognition*, CVPR.

Schölkopf, B., Smola, A., and Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput.*, pages 1299–1319.

Schuldt, C., Laptev, I., and Caputo, B. (2004). Recognizing human actions: a local svm approach. In *Proceedings of the International Conference Pattern Recognition*, CVPR, pages 32–36.

Scovanner, P., Ali, S., and Shah, M. (2007). A 3-dimensional SIFT descriptor and its application to action recognition. In *Proceedings of the international conference on Multimedia*, pages 357–360.

Serre, T., Wolf, L., and Poggio, T. (2005). Object recognition with features inspired by visual cortex. In *Proceedings of the International Conference Pattern Recognition*, CVPR, pages 994–1000.

Shao, L. and Mattivi, R. (2010). Feature detector and descriptor evaluation in human action recognition. In *Proceedings of the ACM International Conference on Image and Video Retrieval*, pages 477–484.

Shechtman, E. and Irani, M. (2007). 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*, pages 2045–2056.

Shivakumar, N. and Garcia-Molina, H. (1999). Finding near-replicas of documents on the web. In *The World Wide Web and Databases*, volume 1590 of *Lecture Notes in Computer Science*, pages 204–212. Springer Berlin Heidelberg.

Singh, S., Ren, W., and Singh, M. (2009). A novel approach to spatio-temporal video analysis and retrieval. In *Processing of the Computer Vision/Computer Graphics CollaborationTechniques*, pages 106–115. Lecture Notes in Computer Science.

Sivic, J., Everingham, M., and Zisserman, A. (2005). Person spotting: Video shot retrieval for face sets. In *Proceedings of the 4th International Conference on Image and Video Retrieval*, CIVR'05, pages 226–236.

Smeaton, A., Over, P., and Kraaij, W. (2009). High-level feature detection from video in TRECVid: A 5-year retrospective of achievements. In *Multimedia Content Analysis*, Signals and Communication Technology, pages 1–24.

Smith, L. (2001). A tutorial on principal components analysis. Technical report, USA: Cornell University.

Souvenir, R. and Pless, R. (2005). Manifold clustering. In *Proceedings of the IEEE International Conference on Computer Vision*, ICCV.

Spearman, C. (1904). "General intelligence," objectively determined and measured. *American Journal of Psychology*, pages 201–293.

Stehman, S. V. (1997). Selecting and interpreting measures of thematic classification accuracy. *Remote Sensing of Environment*, pages 77 – 89.

Stein, G. (1999). Tracking from multiple view points: Self-calibration of space and time. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, CVPR, pages 521–527.

Sun, C., Junejo, I. N., and Foroosh, H. (2011). Action recognition using rank-1 approximation of joint self-similarity volume. In *Proceedings of the IEEE International Conference on Computer Vision*, ICCV, pages 1007–1012.

Swets, D. L. and Weng, J. (1996). Using discriminant eigenfeatures for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 831–836.

Tenenbaum, J. B. (1998). Mapping a manifold of perceptual observations. In *Advances in Neural Information Processing Systems 10*, pages 682–688. MIT Press.

Tenenbaum, J. B., de Silva, V., and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, pages 2319–2323.

Torr, P. and Zisserman, A. (2000). Feature based methods for structure and motion estimation. In *Vision Algorithms: Theory and Practice*, pages 278–294. Springer Berlin Heidelberg.

Uz, K., Vetterli, M., and LeGall, D. (1991). Interpolative multiresolution coding of advance television with compatible subchannels. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 86–99.

van der Maaten, L., Postma, E. O., and van den Herik, H. J. (2008). Dimensionality reduction: A comparative review. Technical report, MICC, Maastricht University.

van Gemert, J. C., Geusebroek, J.-M., Veenman, C. J., and Smeulders, A. W. M. (2008). Kernel codebooks for scene categorization. In *Proceedings of the 4th European Conference on Computer Vision*, ECCV, pages 696–709.

Vaughan, L. (2004). New measurements for search engine evaluation proposed and tested. *Information Processing and Management*, pages 677 – 691.

Vedaldi, A. and Fulkerson, B. (2010). Vlfeat: an open and portable library of computer vision algorithms. In *Proceedings of the international conference on Multimedia*, MM, pages 1469–1472.

Vedaldi, A. and Zisserman, A. (2012). Efficient additive kernels via explicit feature maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 480–492.

Wang, H., Kläser, A., Schmid, C., and Liu, C.-L. (2013). Dense trajectories and motion boundary descriptors for action recognition. *International Journal of Computer Vision*, pages 60–79.

Wang, H., Ullah, M. M., Kläser, A., Laptev, I., and Schmid, C. (2009). Evaluation of local spatio-temporal features for action recognition. In *Proceedings of British Machine Vision Conference*, BMVC, pages 127–127.

Wang, J., Chen, Z., and Wu, Y. (2011). Action recognition with multiscale spatio-temporal contexts. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, CVPR, pages 3185–3192.

Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., and Gong, Y. (2010). Locality-constrained linear coding for image classification. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, CVPR, pages 3360–3367.

Wang, R., Shan, S., Chen, X., and Gao, W. (2008). Manifold-manifold distance with application to face recognition based on image set. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, CVPR.

Wedge, D., Kovesi, P., and Huynh, D. (2005). Trajectory based video sequence synchronization. In *Proceedings of Digital Image Computing: Techniques and Applications*, pages 13–13.

Weinland, D., Özuysal, M., and Fua, P. (2010). Making action recognition robust to occlusions and viewpoint changes. In *Proceedings of the European Conference on Computer Vision*, ECCV, pages 635–648.

Willems, G., Tuytelaars, T., and Gool, L. (2008). An efficient dense and scale-invariant spatio-temporal interest point detector. In *Proceedings of the 10th European Conference on Computer Vision*, ECCV, pages 650–663.

Wold, S., Esbensen, K., and Geladi, P. (1987). Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*.

Wong, S.-F. and Cipolla, R. (2007). Extracting spatiotemporal interest points using global information. In *Proceedings of IEEE International Conference on Computer Vision*, ICCV, pages 1–8.

Wu, S., Oreifej, O., and Shah, M. (2011). Action recognition in videos acquired by a moving camera using motion decomposition of lagrangian particle trajectories. In *Proceedings of the IEEE International Conference on Computer Vision*, ICCV, pages 1419–1426.

Xia, L. and Aggarwal, J. (2013). Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera. In *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR, pages 2834–2841.

Xiao, W., Wang, B., Liu, Y., Bao, W., and Zhang, M. (2014). Context-aware and locality-constrained coding for image categorization. *The Scientific World Journal*, pages 1–14.

Xu, G. and Zhang, Z. (1996). *Epipolar Geometry in Stereo, Motion and Object Recognition: A Unified Approach*. Computational Imaging and Vision. Springer.

Yamaguchi, O., Fukui, K., and Maeda, K. (1998). Face recognition using temporal image sequence. In *Proceedings of International Conference on Face and Gesture Recognition*.

Yang, J., Yu, K., Gong, Y., and Huang, T. (2009). Linear spatial pyramid matching using sparse coding for image classification. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, CVPR, pages 1794–1801.

Yang, J., Zhang, D., Yang, J. Y., and Niu, B. (2007). Globally maximizing, locally minimizing: Unsupervised discriminant projection with applications to face and palm biometrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Yang, L. (2005). Building k-edge-connected neighborhood graph for distance-based data projection. *Pattern Recognition Letters*, 26(13):2015 – 2021.

Yang, M., Zhang, L., Yang, J., and Zhang, D. (2011). Robust sparse coding for face recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, CVPR, pages 625–632.

Yao, A., Gall, J., and Gool, L. V. (2010). A Hough transform-based voting framework for action recognition. In *Proceedings of the International Conference Pattern Recognition*, CVPR, pages 2061–2068.

Yeffet, L. and Wolf, L. (2009). Local trinary patterns for human action recognition. In *Proceedings of IEEE International Conference on Computer Vision*, ICCV, pages 492–497.

Yuan, C., Li, X., Hu, W., Ling, H., and Maybank, S. (2013). 3D R transform on spatio-temporal interest points for action recognition. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, CVPR, pages 724–730.

Yusuf Aytar, M. S. a. L. (2008). Utilizing semantic word similarity measures for video retrieval. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, CVPR.

Zaslavskiy, M., Bach, F., and Vert, J. P. (2010). Many-to-many graph matching: a continuous relaxation approach. In *Proceedings of European Conference on Machine Learning and Knowledge Discovery in Databases*.

Zeng, M., Yang, T., Li, Y., Meng, Q., Liu, J., and Han, T. (2011). Finding regions of interest based on scale-space keypoint detection. In *International conference on Computer Science and Education Applications*, Communications in Computer and Information Science, pages 428–435. Springer Berlin Heidelberg.

Zhang, Z., Deriche, R., Faugeras, O., and Luong, Q.-T. (1995). A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, pages 87 – 119.

Zhen, X. (2013). *Feature Extraction and Representation for Human Action Recognition*. PhD thesis, University of Sheffield, Sheffield, UK.

Zhou, F. and de la Torre, F. (2012). Factorized graph matching. In *Proceedings of the IEEE International Conference on Computer Vision*, ICCV.

Zoghlami, I., Faugeras, O., and Deriche, R. (1997). Using geometric corners to build a 2D mosaic from a set of image. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, CVPR, pages 420–428.