Video Sequence Alignment and its Applications in Video Searching and Retrieval

Manal Alghamdi and Yoshihiko Gotoh
Department of Computer Science
University of Sheffield

\{egp08mag, y.gotoh\}@sheffield.ac.uk

March 2012
Abstract

Analysing video streams is a fundamental task in many applications. It demands the extraction of different levels of feature that represent the events. This research investigates the works done in the video searching and retrieval field and it centres around the issues combined with this task. Some of these issues relate to finding a good representation for the video content and to defining the similarity concept between two representations.

The presented report provides a brief overview of my progress in the first year of my PhD degree and involves a comprehensive review of the literature, an outline of my research and a space-time extension of the 2D Scale-Invariant Feature Transform (SIFT), originally applied to the volumetric images in 2D. This extension proved its ability to represent the space-time events in the video and is well suited to classifying the human actions. An experiment was conducted for the action recognition task using different datasets and, compared to the other approaches, the presented approach among the state-of-the-art.
Contents

List of Figures v
List of Tables vii
Glossary ix

1 Introduction 1
  1.1 Motivation ......................................................... 3
  1.2 Research Statement ............................................. 4
  1.3 Contributions ................................................... 5
  1.4 Datasets .......................................................... 6
    1.4.1 KTH .......................................................... 6
    1.4.2 UCF Sports .................................................. 6
    1.4.3 Rushes Video ............................................... 8
    1.4.4 Instance Search Dataset .................................... 8
  1.5 The Outline of the Report ....................................... 9

2 Related Work 11
  2.1 Interest Points Features Approaches ............................... 11
    2.1.1 Spatio-Temporal Features ................................... 11
    2.1.2 Scale-Invariant Feature Transform (or SIFT) .................. 12
      2.1.2.1 Spatial SIFT ........................................... 12
      2.1.2.2 3D-SIFT Extension .................................... 13
    2.1.3 SIFT Feature in Video Processing ............................. 15
  2.2 Feature Learning Algorithms ..................................... 16
    2.2.1 Vector Quantization ........................................ 17
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2.2</td>
<td>Sparse Coding</td>
<td>18</td>
</tr>
<tr>
<td>2.2.3</td>
<td>Sparse Coding in Video Processing</td>
<td>20</td>
</tr>
<tr>
<td>2.3</td>
<td>Dimensionality Reduction</td>
<td>21</td>
</tr>
<tr>
<td>2.3.1</td>
<td>Traditional Linear Techniques</td>
<td>21</td>
</tr>
<tr>
<td>2.3.1.1</td>
<td>Principle Component Analysis (PCA)</td>
<td>22</td>
</tr>
<tr>
<td>2.3.2</td>
<td>Non-linear Techniques</td>
<td>23</td>
</tr>
<tr>
<td>2.3.2.1</td>
<td>Global Non-linear Techniques</td>
<td>23</td>
</tr>
<tr>
<td>2.3.2.2</td>
<td>Local Non-linear Techniques</td>
<td>26</td>
</tr>
<tr>
<td>2.3.2.3</td>
<td>Isomap in Video Processing</td>
<td>27</td>
</tr>
<tr>
<td>2.4</td>
<td>Video Searching and Retrieval</td>
<td>29</td>
</tr>
<tr>
<td>2.4.1</td>
<td>Video Similarity Approaches</td>
<td>29</td>
</tr>
<tr>
<td>2.4.2</td>
<td>Repetitive Content Approaches</td>
<td>29</td>
</tr>
<tr>
<td>2.4.3</td>
<td>Instance Search (INS)</td>
<td>30</td>
</tr>
<tr>
<td>3</td>
<td>Contribution 1: Spatio-temporal SIFT</td>
<td>33</td>
</tr>
<tr>
<td>3.1</td>
<td>ST-SIFT Extension</td>
<td>33</td>
</tr>
<tr>
<td>3.1.1</td>
<td>Spatio-temporal Difference of Gaussian Pyramid</td>
<td>34</td>
</tr>
<tr>
<td>3.1.2</td>
<td>Interest Points Detection</td>
<td>36</td>
</tr>
<tr>
<td>3.2</td>
<td>ST-SIFT Action Recognition Framework</td>
<td>37</td>
</tr>
<tr>
<td>3.2.1</td>
<td>Experimental Setup</td>
<td>38</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Results</td>
<td>38</td>
</tr>
<tr>
<td>4</td>
<td>Outlines for Contributions 2-4</td>
<td>43</td>
</tr>
<tr>
<td>4.1</td>
<td>Contributions 2: Sparse Coding and Manifold Representation</td>
<td>43</td>
</tr>
<tr>
<td>4.2</td>
<td>Contributions 3: Video Clips Similarity and Classification</td>
<td>45</td>
</tr>
<tr>
<td>4.3</td>
<td>Contributions 4: Instance Search (or INS)</td>
<td>45</td>
</tr>
<tr>
<td>5</td>
<td>Achievements and Future Plan</td>
<td>47</td>
</tr>
<tr>
<td>5.1</td>
<td>Achievements</td>
<td>47</td>
</tr>
<tr>
<td>5.1.1</td>
<td>DDP Summary</td>
<td>47</td>
</tr>
<tr>
<td>5.1.2</td>
<td>Workshop/ Seminar Attendance</td>
<td>48</td>
</tr>
<tr>
<td>5.1.3</td>
<td>Coming Workshop/ Seminar</td>
<td>48</td>
</tr>
<tr>
<td>5.1.4</td>
<td>Experiments</td>
<td>48</td>
</tr>
<tr>
<td>5.2</td>
<td>Next Six-months Plan</td>
<td>49</td>
</tr>
</tbody>
</table>
List of Figures

1.1 Example of the actions in the KTH dataset (Schuldt et al., 2004) . . . . 7
1.2 Some examples of the actions provided by the UCF sports dataset (Ro-

   driguez et al., 2008) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7
1.3 Some frames from the BBC rushes video collection . . . . . . . . . . . . 8
2.1 The two-dimensional space of codewords (Swilem et al., 2009). . . . . . 18
2.2 The representation of an image by a limited number of basis functions

   \( a_i \) using sparse coding (Olshausen and Fieldt, 1997). . . . . . . . . 19
2.3 The Isomap interpretation for the swiss roll dataset (Tenenbaum et al.

   2000). . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 25
2.4 Steps for Isomap manifold learning (Chantamunee and Gotoh, 2010) . . 26
2.5 The three steps of the LLE algorithm (Roweis and Saul, 2000). . . . . . 27
3.1 Comparison between the interest points extracted with the 2D-SIFT

   (left) and the proposed ST-SIFT (right). The 2D-SIFT defines the spa-

   tial points only from the moving objects and the background, while the

   ST-SIFT defines the spatial points from different scales that have motion

   information. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 34
3.2 The video pyramids. Each level is a spatially and temporally down-

   sampled version of the previous one, convolved with the 3D Gaussian

   to create the Gaussian pyramid. The DOG is then constructed by sub-

   tracting the adjacent Gaussian scales. . . . . . . . . . . . . . . . . . . . . 35
LIST OF FIGURES

3.3 The mapping strategy between levels in the spatio-temporal Gaussian pyramid with 3 levels. Each pixel in the lower levels corresponds to a pixel in $G_0(2^n)$, e.g. $G_1(3)$ is mapped to $G_0(6)$ and $G_2(1)$ is mapped to $G_0(4)$. 36

3.4 The video volume and the generated frames along different directions. 37

4.1 System framework for video shots representation. 44

4.2 A framework for the two applications: 1) video clips similarity and classifications within the same video file; 2) instance search task for specific query clip within the video stream. 46
# List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Action recognition results. (Top) Confusion matrix for the KTH dataset</td>
<td>39</td>
</tr>
<tr>
<td></td>
<td>with 90.74% recognition rate. (Bottom) Confusion matrix for the UCF dataset</td>
<td></td>
</tr>
<tr>
<td></td>
<td>with 80.56% recognition rate.</td>
<td></td>
</tr>
<tr>
<td>3.2</td>
<td>Comparative results on the UCF sports and KTH datasets</td>
<td>40</td>
</tr>
<tr>
<td>3.3</td>
<td>Comparative results on the KTH and UCF sports datasets</td>
<td>40</td>
</tr>
<tr>
<td>5.1</td>
<td>DDP module</td>
<td>48</td>
</tr>
<tr>
<td>5.2</td>
<td>Work plan for the next six months</td>
<td>49</td>
</tr>
<tr>
<td>5.3</td>
<td>Work plan for the next two years</td>
<td>50</td>
</tr>
</tbody>
</table>
## Glossary

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSER</td>
<td>Maximally Stable Extremal Regions</td>
</tr>
<tr>
<td>HAR</td>
<td>Harris-Laplace detector</td>
</tr>
<tr>
<td>HES</td>
<td>Hessian-Laplace detector</td>
</tr>
<tr>
<td>HOF</td>
<td>Histogram of Optical Flow</td>
</tr>
<tr>
<td>HOG</td>
<td>Histogram of Gradients</td>
</tr>
<tr>
<td>GLOH</td>
<td>Gradient Location and Orientation Histogram</td>
</tr>
<tr>
<td>SPREF</td>
<td>Spatio-temporal Regularity Flow</td>
</tr>
<tr>
<td>SIFT</td>
<td>Scale-Invariant Feature Transform</td>
</tr>
<tr>
<td>ST-SIFT</td>
<td>Spatio-temporal Scale-Invariant Feature Transform</td>
</tr>
<tr>
<td>DoG</td>
<td>Difference of Gaussian</td>
</tr>
<tr>
<td>TRECVID</td>
<td>TREC video retrieval evaluation</td>
</tr>
<tr>
<td>INS</td>
<td>Instance Search</td>
</tr>
<tr>
<td>LBP</td>
<td>Local Binary Patterns</td>
</tr>
<tr>
<td>BoF</td>
<td>Bag of Features</td>
</tr>
<tr>
<td>SVM</td>
<td>Support Vector Machine</td>
</tr>
<tr>
<td>LOOCV</td>
<td>Leave-One-Out-Cross-Validation</td>
</tr>
<tr>
<td>PCA</td>
<td>Principle Component Analysis</td>
</tr>
<tr>
<td>CFMT</td>
<td>Compact Fourier Mellin Transform</td>
</tr>
<tr>
<td>PDH</td>
<td>Pyramid Density Histogram</td>
</tr>
<tr>
<td>NFT</td>
<td>Nearest Feature Trajectory</td>
</tr>
<tr>
<td>MDS</td>
<td>Multidimensional Scaling</td>
</tr>
<tr>
<td>Isomap</td>
<td>Isometric Feature Mapping</td>
</tr>
<tr>
<td>LLE</td>
<td>Locally Linear Embedding</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

Video search engines, such as Google Videos and YouTube, have become the most frequently visited websites recently. Search normally depends on annotations such as titles, tags, and descriptions created manually. One potential problem is that annotations do not always reflect the actual content of a video. In addition, manual annotation is a comprehensive and tedious process that requires human intervention. There are three types of information that can be used to represent the video content: visual content, audio information and text information. Most of current works rely on the visual information or combination of visual and text information. Some approaches used the audio information by converting it to text information using automatic speech recognition (ASR) engines. For the most tasks, best results were obtained by employing visual contents only.

In this research I aim to explore approaches to indexing, searching and retrieving videos given a video stream as a query. This should be accomplished by analysing the visual contents rather than text annotations. I investigate video representation that is invariant to various transformation, environment and circumstances. I study techniques to represent video events in space-time domain. The developed techniques will be verified by applications within the video searching tasks. Several applications in the video searching and retrieval field will be developed including an extension of the instance search task from the TREC video retrieval evaluation (TRECVID) competition. Multiple datasets including the rushes video (also known as pre-production video) will be used for technical development in video similarity search.

The work can be summarised with the following two stages:
1. INTRODUCTION

• Firstly the scale-invariant feature transform (or SIFT) algorithm is extended in the time domain. This provides a representation for video data with invariant descriptors to scale, location, orientation and time changes. Using the features learning techniques such as sparse coding, a dictionary of the extracted features can be created to convert the low-level features to mid-level features with richer representation. The final step is to apply a non-linear dimensionality reduction technique such as Isomap to discover the underlying structure of video data and to align them in a lower dimensional space.

• Secondly similarity of video is measured using the first stage outcomes with applications include alignment of video streams and video instance search given a video clip as a query.

I spent the past six-months to developing the spatio-temporal SIFT (or ST-SIFT) approach to detecting interest points that have significant local variations in both spatial and temporal domains. In addition, presented features are invariant to scale, location, orientation changes in spatial and temporal domains. With this work, I tackled two important issues that have not been addressed in the previous extensions made for SIFT. One is the transformation of video signal into 3D spatio-temporal pyramids that deal with the 2D space and additional time domain separately. 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.

Representation produced by the ST-SIFT generates a low-level features for a video stream. Sparse coding is applied to create a model of mid-level features with reduced high-order redundancy. After that, the generated model is transferred by a dimensionality reduction technique to a trajectory in the lower dimensional space. Developing technical framework will be evaluated using two different tasks within the video searching and retrieval applications. The first is video clips categorization, i.e. classification of clips to semantically meaningful classes based on generated representations. The second is inspired by the TRECVID tasks. The instance search (INS) task aims to locate different types of image query in a collection of test video clips. We will adapt this idea to an application of searching for desired clips within a video collection.
1.1 Motivation

The presented work can be a starting point for different real world applications in the video domain. Providing a tool for video sequences alignment can be a basic step in several tasks such as indexing, parsing, similarity, action recognition, searching and retrieval. Considering a complex data sequence such as the rushes video in the development will give the confidence that the presented approach is able to deal with real world data such as surveillance, news and movies.

1.1 Motivation

With the rapid progress in video processing techniques, representation of video contents is still an open and challenging task. A number of video fingerprinting techniques have been developed for indexing and retrieval which rely on one level of features. However, applying conventional approaches on more complicated data such as rushes video with multiple retakes may generate weak fingerprints with similar representation for different streams. A combination of low-level features, coding techniques and manifold learning will provide a better fingerprint for such data that can also be used in various applications. Combining these approaches will have the following advantages:

- Low-level features will capture the salient properties of the visual patterns under different circumstances.

- The coding technique will provide a mid-level representation saving both processing time and space for the visual descriptors, which will lead to a more robust representation with a carefully constructed dictionary.

- The various representations provide a tool for modelling video sequence by mapping each frame to point in the low-dimensional space. This will estimate the underlying structure of complex high-dimensional data.

An important question that raises from visual similarity applications, is the meaning of semantic analysis in representing visual contents?. How to measure the similarity of two visual streams?. This makes research in this field challenging and exciting at the same time.

In this research, an attempt will be made to define visual representations for video contents in the spatio-temporal domain and to measure video similarity. Having a
1. INTRODUCTION

definitive concept for representation and similarity are the major steps in various applications. Potential benefits from video content analysis apply not only to video searching and retrieval tasks but also to other areas such as human activity understanding, video summarisation and copy detection. The approach can be extended to represent various types of video contents for different applications such as sports video analysis and video surveillance.

1.2 Research Statement

This research aims to solve issues relating to video processing tasks. It will focus on two main problems, video representation and video similarity. The first stage builds on multiple methodologies that have been used separately for visual representation of data. A novel approach will be investigated that extends the state-of-the-art image processing technique to accommodate temporal information. The second stage focuses on the video similarity task using representation of video with various level of features developed in the first stage. They are outlined as follows:

- **Stage 1:** Video contents, invariant to scale, location, orientation changes are robustly and accurately described, in the spatial and the temporal domains.
  - **Step 1:** Develop a 3D-SIFT (2D space + time) that are able to extract a highly distinctive features, robust against temporal and spatial changes in video.
  - **Step 2:** Explore most suitable features representation for a video stream in manifold using a dimensionality reduction technique.
  - **Step 3:** Align multiple video streams to demonstrate the important transitions within the sequence. This can be a baseline for various applications in visual analysis such as event detection, video similarity, content copy detection and repetitive content detection.

- **Stage 2:** The similarity of videos is explored at various levels. From the trajectory alignment, several applications can be defined using different visual units such as shots, clips or the entire sequence.
1.3 Contributions

Research and development in this thesis cover important problems in the video processing domain. The major contributions will be as follows:

- The first contribution is to develop an approach to detecting and describing interest points that have significant local variations in both spatial and temporal domains and that are invariant to scale, location, orientation changes in spatial and temporal domains. Therefore, the spatial 2D-SIFT is extended to a new version with better ability to identify time-invariant visual features from a video sequence and represent this information using low-level descriptors. The derived representation was evaluated in action recognition tasks and comparable to the state-of-the-art results.

- The second contribution synchronises and aligns repetitive contents in video stream. This approach combines ST-SIFT for interest point extraction, sparse coding for feature learning and Isomap for dimensionality reduction and manifold representation. The sparse coding uses low-level features to create a global representation that reflect the video information. Then a dimensionality reduction technique is applied to reduce the vast number of features and the number of dimensions required to represent the video stream.

- The third contribution is to explore a concept of video similarity at various levels. A video similarity and classification application will be built to demonstrate the ability of developed approaches. There are two aspects that will be covered: measuring similarity and categorizing video clips into classes.

- Lastly contribution is inspired by the TRECVID instance search task. The task involves locating different queries of still image that contains a person, object, or place entity in video clips. This task will be extended to be search video clips within a given video data collection to retrieve a ranked list of the relevant video clips. This will give confidence that the developed approach can be applied to more advance applications.
1. INTRODUCTION

1.4 Datasets

We will evaluate the proposed approach using four different datasets; the KTH dataset \cite{schuldt2004recognizing}, the UCF sports dataset \cite{rodriguez2008evaluation}, the rushes video and the instance search dataset from TREC video. The KTH is the standard dataset used by most human activity analysis researches. The UCF sports dataset is more challenging and realistic with a wide range of scenes and viewpoints in complex environments. To raise the challenge further, the rushes video collection will be used, consisting of a large amount of raw material with highly redundant contents.

1.4.1 KTH

The KTH dataset was created by Schuldt et al. \cite{schuldt2004recognizing} as a collection of human actions under controlled actions. It contains six different human activities, including walking, jogging, running, hand-waiving, boxing and hand-clapping. Each action is performed by 25 people in four different scenarios outdoors s1, outdoors with scale variation s2, outdoors with different clothes s3 and indoors s4 as shown in Figure 1.1. It consists of 600 videos sampled at 25fps (frame per seconds) with the frame size of 160 x 120 pixels. The background is homogeneous and static for most cases, containing single identifiable object. Therefore, it is considered as a standard dataset for action detection and recognition approaches.

1.4.2 UCF Sports

Since the KTH dataset is relatively simple, other datasets with more complex and challenging content with different viewpoints and background are considered. The UCF sports dataset was collected by Rodriguez et al. \cite{rodriguez2008evaluation} from sport broadcasting videos. It contains ten human actions but the pole vaulting action is not publicly available. Therefore, we apply our approach to the nine available actions consisting of diving, golf swinging, kicking, lifting, horseback riding, running, skating, swinging and walking. There are approximately 150 videos showing a large intra-class variability (cf. Figure 1.2). Unlike other action datasets, the UCF sports dataset contains moving background and several objects. Therefore, it is usually applied to prove the approach ability to detect, represent and recognize video events.
1.4 Datasets

Figure 1.1: Example of the actions in the KTH dataset \cite{Schuldt2004}.

Figure 1.2: Some examples of the actions provided by the UCF sports dataset \cite{Rodriguez2008}.
1. INTRODUCTION

1.4.3 Rushes Video

Rushes video is a pre-production collection consisting of 615 video clips. They are raw footages used to produce TV programs. Unlike other types of video datasets, rushes contain noisy contents such as clapper bars, color bars and empty white shots. It also contains repetitive contents from retakes for same scenes caused by, e.g., actors mistakes or technical failures. Figure 1.3 presents some frames from the BBC rushes video dataset. Most researches using this dataset are part of the TRECVID rushes summarisation task. Other relate to indexing, parsing or skimming.

1.4.4 Instance Search Dataset

Yearly, different testing video and query images are released to the participants for the INS task. In TRECVID 2011, the testing data was produced form the rushes collection. They automatically decomposed each video in the dataset into short and equally length clips with different names from the original video file. There were a total number of 20,982 test video clips and 25 image test queries. Some image transformations were also applied to random test clips. The task includes recurring queries with people, location and objects in the rushes.
1.5 The Outline of the Report

The report is organised as follows: Chapter 2 reviews previous researches in the related areas. Chapter 3 presents the ST-SIFT algorithm to detect interest points in videos as well as the experiment results within human action recognition framework. The research outline is provided in chapter 4. Chapter 5 demonstrates two applications evaluation for the proposed work. Chapter 6 summarises the achievements during the past six-months, a work plan for the next six-months and for the remaining two years of my PhD. Finally, the report ends with conclusions.
1. INTRODUCTION
Chapter 2

Related Work

In this Chapter, a literature survey from recent researches in the related field is provided. In particular, image descriptors, features learning and dimensionality reduction techniques are discussed.

2.1 Interest Points Features Approaches

This work provides a spatio-temporal extension of the 2D SIFT approach, which are invariant to scale, location and orientation changes in spatial and temporal domains. This section reviews the representative publications in this field.

2.1.1 Spatio-Temporal Features

Local features have received a lot of attention in video processing applications. These features have been extended to take into account the spatio-temporal nature of video data. Laptev and Lindeberg (2003) extended the well-known Harris-Laplace detector in spatial domain to the spatio-temporal domain. The detected points represent the regions with significant variations in both spatial and temporal domains.

Schuldt et al. (2004) build their work on the detectors presented by Laptev and Lindeberg (2003). They provide a video representation from the detected interest point combined with SVM classification schema for action recognition. Their experiment proved that their local features were not sufficient enough to represent the actions and they suffer from the scale and speed of the actions sensitivity.
2. RELATED WORK

Dollar et al. (2005) presented an approach for behavior recognition motivated by the ideas of Laptev and Lindeberg (2003) and Schuldt et al. (2004). They extracted the local maxima from space and time domains based on the responses of the Gaussian filter convolved with a pair of 1D Gabor filter. Their features are not scale invariance and sensitive to illumination changes.

2.1.2 Scale-Invariant Feature Transform (or SIFT)

In the following sections, we will review the original algorithm of the 2D SIFT, followed by extensions accomplished previously.

2.1.2.1 Spatial SIFT

The idea of the 2D SIFT detector is to map a special content of the image to coordinates of scale, location and orientation invariant features. This is achieved using a scale-space kernel function such as Gaussian, which is a continuous function that captures stable features in different scales. The Gaussian function on two points $x$ and $y$ can be defined as:

$$G(x, y, \sigma) = \frac{1}{2\pi \sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$  \hspace{1cm} (2.1)

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

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y, \sigma)$$  \hspace{1cm} (2.2)

where $*$ is the convolution operation. After that, defining stable locations in the scale-space $D(x, y, \sigma)$ can be achieved by the convolution of an input image with the difference of the Gaussian (DoG) functions:

$$D(x, y, \sigma) = (G(x, y, K\sigma) - G(x, y, \sigma)) * I(x, y, \sigma)$$  \hspace{1cm} (2.3)

$$= L(x, y, K\sigma) - L(x, y, \sigma)$$  \hspace{1cm} (2.4)

where the DoG function is the difference between two neighbouring scales with a $K$-constant distance. Finally, defining the maxima and the minima of $D(x, y, \sigma)$ gives the scale invariant points in the scale-space.
2.1 Interest Points Features Approaches

2.1.2.2 3D-SIFT Extension

The extension works for the 2D SIFT can be categorised into three groups. The first one extends the descriptor part only combined 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:

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

(2.5)

where \( L_x, L_y \) and \( L_t \) are the pixel differences approximated by:

\[ L_x = L(x + 1, y, t) - L(x - 1, y, t) \]  

(2.6)

\[ L_y = L(x, y + 1, t) - L(x, y - 1, t) \]  

(2.7)

\[ L_t = L(x, y, t + 1) - L(x, y, t - 1) \]  

(2.8)

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

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

(2.9)

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

(2.10)

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

\[
\begin{bmatrix}
\cos\theta \cos\phi & -\sin\theta & -\cos\theta \sin\phi \\
\sin\theta \cos\phi & \cos\theta & -\sin\theta \sin\phi \\
\sin\phi & 0 & \cos\phi \\
\end{bmatrix}
\]  

(2.11)

Finally, to create the descriptor, a factorisation of sub-histograms is computed for every \( 4 \times 4 \times 4 \) sub-regions with three values for each pixel: one magnitude and two orientations.
In the second category, Cheung and Hamarneh (2007) generalised the 128-D SIFT features to n-dimensional space (n-SIFT) with $2^{5n-3}$-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 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 $3^n+1-1$ neighbours 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, where the extrema that has greater magnitude than $T_{DOG}$ is only considered.

For describing 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 provide a full 3D extension where 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 the 3D image $I(x, y, z, \sigma)$ with Gaussian $G(x, y, z, \sigma)$ as:

$$L(x, y, z, \sigma) = G(x, y, z, \sigma) * I(x, y, z)$$

$$= \frac{1}{(\sqrt{2\pi\sigma}^3 e^{\frac{(x^2+y^2+z^2)}{2\sigma^2}}} * I(x, y, z)$$

The DOG is computed to build a pyramid of blurred images and the local extrema is detected across 3D locations and scale as the features. Due to the nature of the medical images that produced massive number of extrema with poor saliency, they applied 3D filtering step 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 $El \in [-\frac{\pi}{2}; \frac{\pi}{2}]$ are computed similar to Scovanner et al. (2007) angles. The third one $Ti \in [-\pi; \pi]$ is known as roll or tilt angle that build based on the value of the first two angles. In other word, 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) treat the spatial and temporal domains separately. The MoSIFT descriptor contains two parts: describing the spatial domain with histogram of gradient (HOG) and the temporal domain with histogram of optical flow (HOF) that captures the moved 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 amount of motion or action. Secondly, similar to the pyramid of Gaussian, a pyramid of optical flow is constructed for each Gaussian pyramid. Then, local extrema is detected from the DOG pyramid if it contains motion information in the optical flow pyramid. In determining a descriptor, factorisation histogram is used for each kind of features separately with one difference in the dominant orientation. The optical flow does not involve orientation invariant. At the end, a single descriptor is created for both HOG and HOF features with 256 dimensions (128 SIFT + 128 optical flow).

### 2.1.3 SIFT Feature in Video Processing

Using the SIFT algorithm generates a tremendous amount of features for a single image, where each feature is describe by a 128-dimension vector (Sarkar et al., 2008). Therefore, a matching step in the visual similarity and retrieval tasks will suffer from processing speed issues. Various solutions have been explored to reduce the number of features extracted from each image and to reduce the comparison cost and time. A bag of words (BoW) model and dimensionality reduction techniques have produces good results with high-performance.

The BoW concept has been widely used in the text processing area (Chiu and Chen, 2007). It starts by extracting words from each document to construct a model, where the dimension of the feature vector is equal to the number of extracted words. To make it more representative and to reduce the features dimensions, words can be grouped into classes using various techniques. In multimedia processing, the descriptors (words) are extracted from a video or an image by low-level features processing techniques such
2. RELATED WORK

as SIFT. After that, Vector quantisation or K-means clustering algorithms are used to create descriptors categories. This approach has been used in (Xin and Katsaggelos 2010) as a part of the image retrieval framework. They created a bag of visual word for each image and used the Euclidean distance to compare a query BoW with a database with different BoWs. BoW was also used for a video copy detection task (Chiu and Chen 2007). Each video was represented with a BoW model that contained SIFT features extracted from each frame. A Coarse matching approach was applied, which is a histogram based matching technique over a window.

Principle component analysis (PCA) for dimensionality reduction was combined with SIFT features to identify the Near-Duplicate Keyframe (NDK) in a video stream (Zhao et al. 2007). They are the repeated keyframes with slight changes in lightening, motion, camera angles and so on. Identification of the NDKs has been investigated recently for topic detection and tracking and for video copy detection. To solve these challenging tasks, the authors of (Zhao et al. 2007) extracted the features from local interest points and defined a descriptor that captured these features around each of local interest points. Therefore, they applied the SIFT algorithm followed by the PCA to solve the comparison speed issue. Using the extracted patches by SIFT, the gradient image was defined and projected to the output space. After that, the descriptor was produced as the top elements of the projected vector in the eigenspace.

2.2 Feature Learning Algorithms

In visual recognition researches, vector quantisation has presented successful results. However there have been recent efforts to employ more powerful techniques for creating a high-level image representation (Coates and Ng 2011). Sparse coding has been considered as a good alternative to vector quantisation and has produced excellent results in various recognition and classification tasks.

Vector quantisation and sparse coding are considered a feature learning algorithms that contain two steps: the training step and the encoding step. During training, basis functions $D$ should be learned as weights, codebook or dictionary. While in the encoding phase, the learned $D$ is used to map each input vector $x$ to the corresponding feature vector $y$ in the output.
2.2 Feature Learning Algorithms

2.2.1 Vector Quantization

Vector quantization is a lossy compression technique for mapping \( N \)-dimensional data set in the input space \( \mathbb{R}^N \) to a limited number of vectors \( V \) called code vectors or codewords \cite{Kumaraswamy2004}. A collection of \( K \) codewords is known as a codebook where \( V = \{ v_i : i = 1, 2, \ldots, K \} \). This technique follows the concept of a block coding that captures each observation from the input space by specific number of points selected to prevent the loss. Each codeword is associated with the Voronoi region as shown in Figure 2.1, which is the nearest neighbour region and can be described as:

\[
V_i = \left\{ x \in \mathbb{R}^N : \| x - y_i \|_2 \leq \| x - y_j \|_2, \forall j \neq i \right\} \quad (2.14)
\]

A group of the Voronoi region represents the input space as follows:

\[
\bigcup_{i=1}^{K} v_i = \mathbb{R}^N \quad (2.15)
\]

\[
\bigcap_{i=1}^{K} v_i = \emptyset \quad (2.16)
\]

In signal processing, given a signal with a set of feature vectors of size \( M \), \( X = [x_1, \ldots, x_N]^T \in \mathbb{R}^{M \times N} \) and groups of codewords \( V = [v_1, \ldots, v_K]^T \) \cite{Yang2009}. Mathematically vector quantization solves the following problem:

\[
\min_{V} \sum_{n=1}^{N} \min_{k=1\ldots K} \| x_n - v_k \|_2^2 \quad (2.17)
\]

where \( x_n \) is the \( n^{th} \) input vector, \( v_k \) is the \( k^{th} \) element of the codebook \( V \), \( k \) is the centre of the defined codeword and \( \| \cdot \|_2 \) is the \( l_2 \)-norm.

Following the matrix factorization concept, this problem can be solved as:

\[
\min_{\alpha, V} \sum_{n=1}^{N} \| x_n - V \alpha_n \|_2^2 \quad (2.18)
\]

subject to \( \text{Card}(\alpha_n) = 1, |\alpha_n| = 1, \alpha_n \geq 0, \forall n \).
2. RELATED WORK

![Diagram of codewords and vectors]

Figure 2.1: The two-dimensional space of codewords (Swilem et al., 2009).

where $\alpha = [\alpha_1, \ldots, \alpha_N]^T$ is an indicator associated with each codebook, $\text{Card}(\alpha_n) = 1$ is the cardinality restriction to have only one nonzero element in each $\alpha_n$. This optimization involves two steps: the training step considers both $\alpha$ and $V$, and the coding step where the updated $V$ is used to solve the problem with respect to the $\alpha$. The result from this optimisation is the nonzero value in the $\alpha_n$ used to determine the cluster of $x_n$ vector.

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 the higher-order redundancy. Compared with other unsupervised methods for learning such as PCA, sparse coding generates a model of un-orthogonal basis functions that can be adapted easily. Moreover sparse coding is over-complete, where the number of basis functions exceeds the input dimensions. This leads to higher chances in finding better sparse representation for an input signal.

Reducing the high-redundancy in data involves a probabilistic model to define the image architecture (Olshausen and Fieldt, 1997). This is accomplished by linearly
2.2 Feature Learning Algorithms

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 (Figure 2.2). The prominent feature of these events is the sparsity, where an input image can be interpreted with a few basis functions out of a large set.

![Sparse representation](image)

**Figure 2.2:** The representation of an image by a limited number of basis functions $a_i$ using sparse coding (Olshausen and Field, 1997).

Mathematically, let $X = [x_1, \ldots, x_N] \in \mathbb{R}^{M \times N}$ be a matrix of input signal and each column $x_n$ is an $M$-dimensional column vector of extracted features such as SIFT. $D \in \mathbb{R}^{M \times K}$ be a dictionary and each column is an $M$-dimensional basis vector (Sprechmann and Sapiro, 2010). To obtain a sparse representation, the following optimization problem is solved:

$$\min_{\alpha_n} \{\|x_n - D\alpha_n\|_2^2 + \lambda \|\alpha_n\|_1\} \quad (2.19)$$

where $\|\alpha_n\|_1$ is the $l_1$-norm of the weight or coefficients vector $\alpha_n \in \mathbb{R}^K$ that represents the number of non-zero elements, and $\lambda$ is a regularization parameter that maintains the reconstruction error with respect to the sparsity.

This optimization is a convex problem that can be solved in two iterative steps: sparse coding with fixed dictionary $D$ and updating the dictionary with fixed $\alpha$ (Lee, 2010).
2. RELATED WORK

Sparse coding can be derived from vector quantisation by modifying some of the constraints (Yang and Huang 2009). Firstly, relaxing the constraint $\text{Card}(\alpha_n) = 1$ by applying the $l_1$-norm regularization on $\alpha$. This enforces the $\alpha_n$ values to be small nonnegative elements. Secondly, the sign of the $\alpha_n$ is not important. Considering $D^T \leftarrow [D^T, -D^T]$ and $\alpha^T_n \leftarrow [\alpha^T_n, -\alpha^T_n]$, so the constraint $\alpha_n \geq 0$ can be discarded in the sparse coding.

Sparse coding has been widely used in image representation, machine learning and signal processing due to its attractive characteristics (Yang and Huang 2009). Firstly, unlike vector quantisation, spares coding uses less constraints leading to lower reconstruction error. Secondly, sparse representation can efficiently model the salient features of images. Thirdly, statistic research on images has proven the sparse nature of the images.

2.2.3 Sparse Coding in Video Processing

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 upon the low-level features and reflects the image information but not as a high-level. Spatial pyramid matching (SPM) is a mid-level feature extraction technique 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 low-level feature extraction technique such as SIFT followed by coding approach such as vector quantisation and end with pooling methods such as max pooling.

Chiu and Chen (2007) extended the standard SPM by replacing the vector quantisation with sparse a coding as coding technique. 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 sparse coding was applied to transform the low-level features into desirable representation in sparsity. After that, all the sparse 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 an excellence performance by sparse coding compared
with hard quantisation. It was tested with multiple image databases and presented high ability in the classification task.

Good performance by sparse coding in various image classification and recognition researches provide strong motivation to apply it in video applications. However, there is no known work for sparse coding in video similarity and retrieval tasks.

2.3 Dimensionality Reduction

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., 2007). Basically, it is a conversion tool that moves a data set from the high-dimensional space to a suitable output space with reduced dimensionality. The new dimensions should reflect intrinsic dimensionality for the data, which is the least number of parameters that capture the data features.

Dimensionality reduction techniques can be divided into linear and non-linear techniques depending on the nature of input data. Further non-linear techniques can be classified as global or local techniques depending on the mapping procedure. Generally, all different dimensionality reduction techniques concentrate on the following problem:

Given an input matrix $X$ with $M \times N$ dimensions, where $M$ is the number of the row vectors $x_i$ that has $N$ dimensions. The dimensionality reduction techniques aims to define the data set $Y$ with intrinsic dimensions $d < N$ or even $d \ll N$. The intrinsic dimension is the real dimension where input data $X$ lying on or nearby, and that has been embedding in the high-dimension.

2.3.1 Traditional Linear Techniques

There are two known techniques in this group, PCA and Linear Discriminants Analysis (LDA) (Van der Maaten et al., 2007). These techniques expect that the intrinsic dimension of the input data is a linear sup-space of the high-dimension. PCA was produced by Pearson (1901) and since then it has become the base step for dimensionality reduction techniques.
2. RELATED WORK

2.3.1.1 Principle 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 principle components (PCs) (Smith, 2001). It provides a linear transformation of data $X$ in the high-dimensional space into small matrices $B$ and $U$ in the lower dimensional space. These two matrices define the structure and the essential features of the dataset $X$. Exploring the structure of the $B$ matrix shows the object pattern (rows) in $X$, while the structure of the $U$ matrix defines the variable pattern (columns) in $X$.

Given a data set $X = [x_1, x_2, \ldots, x_M] \in \mathbb{R}^{M \times N}$, a matrix of $M$ row vectors with $N$ length, the PCA can be applied as following (Jolliffe, 1986; Smith, 2001):

1. Compute the mean vector as:

$$
\mathbf{x} = \frac{1}{M} \sum_{m=1}^{M} x_m
$$

(2.20)

2. Subtract the mean vector from each vector in the matrix $X$:

$$
\phi_m = x_m - \mathbf{x}
$$

(2.21)

this will generate a new matrix $A$ containing:

$$
A = [\phi_1, \phi_2, \ldots, \phi_M]
$$

(2.22)

3. Using the new matrix $A$, the $N \times N$ covariance matrix can be computed as:

$$
C = \frac{1}{M} \sum_{m=1}^{M} \phi_m \phi_m^T = AA^T
$$

(2.23)

4. Compute the eigenvectors and the eigenvalues for the square matrix $C$ such that:

$$
eigenvalues of C : \lambda_1 > \lambda_2 > \ldots > \lambda_N
$$

$$
eigenvectors of C : \mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_N
$$

where $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_N$ are the basis vectors for the matrix $X$. Therefore, $X$ or $\mathbf{x} - \mathbf{x}$ can be written as combination of the eigenvectors as:

$$
\mathbf{x} - \mathbf{x} = b_1 \mathbf{u}_1 + b_2 \mathbf{u}_2 + \ldots + b_N \mathbf{u}_N = \sum_{i=1}^{N} b_i \mathbf{u}_i
$$

(2.24)
5. The final step is the dimensionality reduction, choose the eigenvectors with the highest eigenvalues as the principle components (PCs) of $X$:

$$x - \bar{x} = \sum_{n=1}^{K} b_n u_n \quad \text{where } K < N$$  \hspace{1cm} (2.25)

6. The input matrix $X$ can be defined after the PCA as:

$$\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_K \end{bmatrix} = \begin{bmatrix} u_1^T \\ u_2^T \\ \vdots \\ u_K^T \end{bmatrix} (x - \bar{x}) = U^T (x - \bar{x})$$  \hspace{1cm} (2.26)

where $b_1, b_2, \ldots, b_K$ is the representation of the $x - \bar{x}$ using the basis functions.

The goal of using the PCA can be summarised the following in three points (Wold and Geladi, 1987). Firstly, it captures the relations between the internal objects. Secondly, it reduces the number of dimensions required to represent these data while maintaining the existing variations in the original structure. Finally, it measures the correlation of the internal variables in the original structure.

### 2.3.2 Non-linear Techniques

These techniques do not consider the linearity of the input space, which provide better chance to deal with input data with complex embedding in a high-dimensional space (Van der Maaten et al., 2007). However, they have not been studied widely. There are two groups of the non-linear techniques: the global techniques that focuses on the global features in input data and the local techniques that focus on the local features of input data.

#### 2.3.2.1 Global Non-linear Techniques

The global techniques for dimensionality reduction retain the global properties of each data point in data set (Van der Maaten et al., 2007). There are various algorithms, however, Multidimensional scaling (MDS) and Isometric Feature Mapping (Isomap) have been the central focuses in the video-processing domain.

- Multidimensional Scaling (MDS)
Multidimensional scaling is a group of statistical methods used to visualise the underlying structure of data points relations using the geometrical representation. Originally, it started from the psychology area to study human perception of similarity and dissimilarity between a pair of entities. After that, the MDS has 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 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 (Young, 1999; Van Deun and Delbeke, 1999), in calculating the Euclidean distance $d_{ij}$ between a pair of points $i$ and $j$ in a given dimension $a$ can be written as follows:

$$d_{ij} = \sqrt{\sum_{a=1}^{m} (x_{ia} - x_{ja})^2} \quad (2.27)$$

Following the Minkowski model, which is a generalisation metric of the Euclidean distance with $p=1$ or 2, equation (2.27) can be written as:

$$d_{ij} = \left[ \sum_{a=1}^{m} |x_{ia} - x_{ja}|^p \right]^{\frac{1}{p}} \quad (2.28)$$

where $x_i$ and $x_j$ are the coordinates of the points $i$ and $j$ respectively, $i, j = 1, 2, \ldots, n$ with $n$ represents the number of objects, $a = 1, 2, \ldots, m$ with $m$ represents the dimensions number and $\frac{1}{p} \leq 1$ is defined by the experimenter. The mapping or the representation quality is tested using the stress functions such as the row stress function (Van der Maaten et al., 2007). It 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:

$$\phi(Y) = \sum_{ij} \left( \|x_i - x_j\| - \|y_i - y_j\| \right)^2 \quad (2.29)$$
where $\|x_i - x_j\|$ is the Euclidean distance between original object in the high-dimensional space and $\|y_i - y_j\|$ is the Euclidean distance between the representative points in the low-dimensional space.

- **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 representative global solution that has the ability to explore the underlying degree of freedom for complicated data such as handwriting characters and biomedical data. It builds on the MDS that depends on the Euclidean distance, which 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). On the other hand, Isomap overcomes this problem by considering the geodesic distance, which is the shortest path or curve that connect two points in connected set. Figure 2.3 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 neighbour based on the Euclidean distance (dashed line) that does not reflect the intrinsic distance in the low-dimensional space as captured by the geodesic distance (solid curve). The second image (centre) presents the neighbourhood graph $G$ constructed by Isomap with $K=7$ nearest neighbours and $N=100$ data points (shortest path is drawn in red line). The final image (right) is the two-dimensional embedding by the Isomap.

![Figure 2.3: The Isomap interpretation for the swiss roll dataset](Tenenbaum et al., 2000)
2. RELATED WORK

The Algorithm for Isomap has three phases (Figure 2.4) (Tenenbaum et al., 2000). Firstly, for each data point \( x_i \), \( k \) nearest neighbours \( x_j \) are detected using the geodesic distance \( d_x(i,j) \). These distances are used to create the weighted graph \( G \), with each node \( x_i \) connected to its neighbour nodes by the \( d_x(i,j) \) edges. Secondly, using the Dijkstra’s algorithm, the shortest path \( d_G(i,j) \) between each pair of datapoint is computed in the graph \( G \) to (over)estimate the geodesic distances \( d_M(i,j) \) 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 in the high-dimensional data space \( X \) in to the intrinsic low-dimensional space \( Y \).

![Figure 2.4: Steps for Isomap manifold learning (Chantamunee and Gotoh, 2010).](image)

2.3.2.2 Local Non-linear 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., 2007). Locally Linear Embedding (LLE) is a recent method that captures a spotlight in various researches.

- **Locally Linear Embedding (LLE)**

Locally Linear Embedding (LLE) is an unsupervised learning algorithm that maps input data from high-dimensional space to its neighbourhood preserving low-dimensional space (Saul and Roweis, 2001). Similar to the MDS and the PCA, the LLE is considered an eigenvector method. LLE captures the local geometry of each point by computing the coefficients from its neighbour points (Roweis and Saul, 2000). These coefficients define the local features of the manifold nearby each point \( x_i \).
Starting with a data set $X \in \mathbb{R}^{M \times N}$ of $M$ row vectors $x_i$ where each of them has $N$-dimensions, LLE algorithm can be defined in three steps as shown in Figure 2.5:

1. Define the nearest neighbourhood points for each data point $x_i$.
2. Define each point $x_i$ as a linear combination weight factor $w_{ij}$ of its neighbour points $x_j$. Measuring the reconstruction error of this weight is define as:

   \[
   \varepsilon (w) = \sum_i \left| x_i - \sum_j w_{ij} x_j \right|^2
   \]  
   \[ (2.30) \]

3. Finally, find the low-dimensional embedding $y_i$ for each $x_i$ using the $w_{ij}$ and minimising the cost function:

   \[
   \phi (y) = \sum_i \left| y_i - \sum_j w_{ij} y_j \right|^2
   \]  
   \[ (2.31) \]

### 2.3.2.3 Isomap in Video Processing

In video sequence analysis, temporal information showing image trajectories have critical influences. Therefore many approaches define the image high-dimensional space
2. RELATED WORK

to represent a video sequence. Due to the vast number of features and the number of dimensions required to represent it, the dimensionality reduction techniques have been used to find a viable solution. In particular, Isomap with its characteristics has been used frequently presenting good performance in various applications that involves video trajectory.

In recent years, different approaches have been investigated with Isomap for several tasks achieving good results. However Isomap is found sensitive to the data continuity. It may not work well for a video with multiple shots (NIE et al., 2010). Therefore, there are some works that employ the standard Isomap with small extensions, combining it with hashing algorithms or doubling it to solve the sensitivity problem.

One team used Isomap as a hashing technique for extracted keyframes (NIE et al., 2010). The most representative keyframe was firstly extracted from each shot in the video. The luminance coefficients of the keyframes were computed as input features. After that, Isomap is applied globally to project the features to a limited number of points in the low dimensional space. A trajectory of points is generated to represent the video structure, and the hash table is define with hashing function based on the distance between each pair of points.

Another group also explored Isomap to produce a representation of video in the low dimensional space (Pless, 2003). These representations could be utilised in various applications such as video segmentation. They extended the standard Isomap to work in the time domain where unordered images were mapped to a trajectory that expressed the video content. The pixels’ intensity was used as the input feature. The distance between each pair of images was used to apply Isomap. Then a spline curve connected the points to exhibit the video structure.

Liu et al. (2008) used the concept of Isomap with clustering algorithms to produce a trajectory of video with various clips. Firstly, the k-nearest neighbour algorithm was used to cluster the high dimensional space. After that, an intra-cluster graph was constructed based on the spatio-temporal relations of data points. In addition, the inter-clustering graph was constructed from representative points in each intra-cluster. These two graphs created the basic-layer and the hyper-layer of the double-layer algorithm. Finally, Isomap was applied to embed data onto the low-dimensional space. In the results, a video with two segments or clips was represented with double trajectories, one for each segment.
2.4 Video Searching and Retrieval

This section provides an overview of important approaches regarding video similarity, searching and retrieval.

2.4.1 Video Similarity Approaches

A number of works have been presented for searching video streams. Sarkar et al. (2008), for example, developed a framework for duplicated or similar video detection tasks. They used SIFT, Compact Fourier Mellin transform (CFMT) and YCbCr as video fingerprinting. To solve the problem of generating a tremendous amount of features for a single video, a hierarchal vocabulary is created, from the extracted features, rather than a flat one that neglected the inter words relationships. A set of $K$ words was created and each words was then divided into another level of $K$ words, generating a $K \times K$ groups of words and so on till the required depth was reached. The similarity between two signatures $X$ and $Y$ was determined by the distance $d(X, Y)$:

$$d(X, Y) = \sum_{i=1}^{K} \left\{ \min_{i \leq j \leq K} \| X(i) - Y(j) \| \right\}$$

(2.32)

where the $\| X(i) - Y(j) \|$ is the distance between the $i^{th}$ frame of $X$ and $j^{th}$ frame of $Y$. In the experiments, they reported that SIFT features detected the similar videos efficiently with superior performance.

Hoi et al. (2003) proposed a framework with two-stage process to define video similarity measurement. The first stage, coarse search, filter out the videos with low similarity by applying the pyramid density histogram (PDH) technique to define a low-level signature mapped to a low-dimensional feature space. While the second stage, fine search, used a nearest feature trajectory (NFT) to compare the query video with the first stage results. In the experimental, they reported a limited performance by retrieving only 85% of the correct matches.

2.4.2 Repetitive Content Approaches

Several works have been presented in analysing video streams with repetitive contents such as the rushes video. Summarisation, retrieval and indexing were the most popular tasks for this kind of development data.
2. RELATED WORK

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

For rushes parsing, [Dumont and Merialdo (2009)] presented a method for detecting the repetitive segments and then parsing the video content into scene and takes. The video content is subsequently decomposed into one second segment and defined a hierarchical clustering from these segments. 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 multimodal space for similarity and retrieval task. 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 defined in the multimodal space by a triple $(N, T, E)$. Then to reduce the complexity of comparing two trajectory, a 3D-solid shape $S$ is defined as representation of the fundamental characteristics of the video. The similarity between two different rushes is defined by a multimodal 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 to related groups. From each cluster, a representative is defined to represent the unit and a summery of the video is generated.

2.4.3 Instance Search (INS)

The TRECVID is an annual conference series started in 2001 and is sponsored by the National Institute of Standards and Technology (NIST). Every year challenging tasks and large collection of testing and query data are provided to research groups to encourage content analysis and information retrieval. The INS task includes searching for specific person, object or place given by still images in a video clip.
2.4 Video Searching and Retrieval

In TRECVID 2011, thirteen groups participated in this task from different universities. Promising results were presented by the NII group (Le et al., 2011) using three different approaches. The three approaches undertake almost the same procedures but with different techniques for representation or for computing the similarity. For the first run, they extracted the 2D-SIFT features from the testing videos to build a large vocabulary tree. Then for each video, a histogram representation is created by projecting the SIFT descriptors into the vocabulary tree. For online searching, the same procedure is applied for the query images and the histogram intersection metric is computed to rank similarity between the images and the video. The second run follows the same baseline but with color SIFT for features extraction. In the final run, multiple combinations of grey and color SIFT were fused and keypoint matching was used for similarity. Similarly, the TNO group (Schavemaker et al., 2011) submitted three approaches contain SURF points in the first one, bag-of-words, global color, face and skin color detection in the second one, and Eigenface descriptors for face recognition and detection in third run.
Chapter 3

Contribution 1: Spatio-temporal SIFT

This chapter briefly introduces the results from the work done in the past six-months. It proposed ST-SIFT algorithm to represent video content with invariant interest points. These points contain sufficient amount of information to describe the performed actions in the video streams. Most of the previous works extended the algorithm spatially to extract the extrema for the 3D images (Cheung and Hamarneh 2007; Allaire et al. 2008; Chen and Hauptmann 2009) or detected 2D interest point and described it with 3D descriptor (Scovanner et al. 2007). Unlike these works, The defined ST-SIFT detect the spatially-distinctive points with sufficient motion information at multi-scales (cf. Figure 3.1). Both detector and descriptor sides are extended spatially and temporally to precisely define the video events under different transformation and circumstances.

3.1 ST-SIFT Extension

To achieve the 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 are vital information in both spatial and temporal domains and used are used to describe the video events. A pseudocode of the algorithm is provided in appendix A.
3. CONTRIBUTION 1: SPATIO-TEMPORAL SIFT

Figure 3.1: Comparison between the interest points extracted with the 2D-SIFT (left) and the proposed ST-SIFT (right). The 2D-SIFT defines the spatial points only from the moving objects and the background, while the ST-SIFT defines the spatial points from different scales that have motion information.

3.1.1 Spatio-temporal Difference of Gaussian Pyramid

The DoG pyramid provides a bandpass version of the original signal [Dorr et al., 2010], which serves a scale-space of the video to 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 firstly introduced by [Üz et al., 1991], where downsampling is performed in both temporal and spatial domains separately. 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.

For a sequence of images $I(t)$ with $W$ by $H$ pixels size, each pixel $(x, y, t)$ at location $(x, y)$ and frame $t$ is denoted as $I(t)(x, y)$. In the first step, the Gaussian pyramid is constructed with $N$ levels where $N$ is calculated from the frame size. Each level is referred to as $G_i(t)$, $0 \leq i \leq N$ where the highest level $G_0$ is the original video signal. This process is adapted in this work to construct a multilevel spatio-temporal Gaussian
and DOG pyramids as shown in Figure 3.2. Initially, the video signal $I(t)$ 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.,

$$L(x, y, t, \sigma, \tau) = G(x, y, t, \sigma, \tau) * I(x, y, t, \sigma, \tau)$$  \hspace{1cm} (3.1)

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

$$G(x, y, t, \sigma, \tau) = \frac{1}{\sqrt{(2\pi)^3 \sigma^4 \tau^2}} e^{-\left(\frac{x^2 + y^2}{2\sigma^2} - \frac{t^2}{2\tau^2}\right)}$$ \hspace{1cm} (3.2)

**Figure 3.2:** 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.

Similar to Lowe (2004), we generate $S + 3$ scales for each of $O$ levels to guarantee that the local extrema detection will cover a complete octave. Then each lower level is produced by spatially and temporally downsampled the Gaussian smoothed signal at scale $\sigma$ in the previous level. This yields a level with lower frame rate and half-sized frames of the previous one (represented by the left side of each box in Figure 3.2). Therefore, a level $G_i(n)$ has $W/2^i$ by $H/2^i$ frame size and match a point in the same time $t$ as $G_0(2^i n)$ (cf. Figure 3.3).

The second step constructs the DOG pyramid. For each level in the Gaussian pyramid, we generate a DOG level with less number of scales by one by subtracting the adjacent scales within the Gaussian level (as shown in the right side of each box in
3. CONTRIBUTION 1: SPATIO-TEMPORAL SIFT

![Diagram showing mapping strategy between levels in the spatio-temporal Gaussian pyramid with 3 levels.](image)

**Figure 3.3:** The mapping strategy between levels in the spatio-temporal Gaussian pyramid with 3 levels. Each pixel in the lower levels corresponds to a pixel in $G_0(2^n)$, e.g. $G_1(3)$ is mapped to $G_0(6)$ and $G_2(1)$ is mapped to $G_0(4)$.

**Figure 3.2.**

$$D(x, y, \sigma, \tau) = (G(x, y, K\sigma, \tau) - G(x, y, \sigma, \tau)) * I(x, y, \sigma, \tau) \quad (3.3)$$

$$= L(x, y, K\sigma, \tau) - L(x, y, \sigma, \tau) \quad (3.4)$$

### 3.1.2 Interest Points Detection

Once the DOG pyramid has been generated, the local extrema (minima/maxima) of the adjacent scales are combined from $xy$, $xt$ 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 axis (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 3.4). The first one is through the spatial axis to create two types of spatial frames cross the $x$ or the $y$ directions. While the second one creates the sequence of frames from the temporal axis combined with one of the $x$ or $y$ spatial axis. Therefore, three axis of each DoG scale volume represent $Space_X$, $Space_Y$ and time dimensions, while 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 union of the results are taken. This allows to introduce tolerance i.e. include points where spatial and temporal extrema may not be at exactly same pixel but nearby each other, and this tolerance is controlled by a threshold value.
3.2 ST-SIFT Action Recognition Framework

Similar to the 2D SIFT [Lowe 2004], the local extrema are detected by comparing each sample point to its eight neighbors in the current scale and the nine neighbors in the scale above and below. This is performed for each level within the DOG pyramid. At the end a filtering step is applied to remove the noisy and edges points.

3.2 ST-SIFT Action Recognition Framework

In this section, the proposed ST-SIFT detector was evaluated in human action recognition task by classifying the presented activity in the given video. The methodology we adopted is the Bag of Features (BoF) model implemented in the ViFeat 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 action classification framework in video data.

The first 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 $\theta$ and $\phi$ angles in the sub-histograms. We used the publicly available code by Scovanner web page, which was slightly different from what was described in [Scovanner et al. 2007]. The code generates a 640-dimensional vector for each interest point.

![Diagram of video volume and generated frames](image-url)
3. CONTRIBUTION 1: SPATIO-TEMPORAL SIFT

The second step is the vocabulary learning, where the generated descriptors from all the interest points are clustering to pre-specified number of visual words. We used the k-means clustering method that implements the Elkan algorithm, which is faster than the standard Lloyd k-means method. The centres of the generated clusters are called ‘visual word’ while the set of these words is known as ‘spatio-temporal word vocabulary’.

Using the computed vocabulary, the third step is 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 known as signature.

Finally, these signatures are used by the Support Vector Machine (SVM) classifier to train representative models for each action. We use a non-linear SVM with a $X^2$-kernel.

3.2.1 Experimental Setup

We employ two different datasets, KTH provided by Schuldt et al. (2004) and UCF sports dataset collected by Rodriguez et al. (2008). Following what has been done in Shao and Mattivi (2010) and Niebles et al. (2008), the KTH dataset is divided into two parts: 16 persons for training and 9 people for testing. While for the UCF sports dataset, the Leave-One-Out-Cross-Validation (LOOCV) training method was used as in the original paper (Rodriguez et al., 2008). For constructing the Gaussian pyramid, the number of scales was set by experiment to 3 scales for each of 4 levels in the KTH dataset, and 3 scales for each of 3 levels in the UCF sports dataset.

For the BoF model, the codebook size is a key-parameter. Following the experiment performed in Shao and Mattivi (2010), the best performance was obtained with a codebook of size 1500 words for both datasets.

For classification, we trained a single SVM classifier for each action using all the samples except the subset left for the testing.

3.2.2 Results

This section demonstrates the performance of the proposed ST-SIFT interest points detector applied to different datasets. As can be seen in the confusion matrices in Table 3.1, the average recognition rates we obtained are 90.74% for the KTH dataset and 80.56% for the UCF sports dataset.
3.2 ST-SIFT Action Recognition Framework

Table 3.1: Action recognition results. (Top) Confusion matrix for the KTH dataset with 90.74% recognition rate. (Bottom) Confusion matrix for the UCF dataset with 80.56% recognition rate.

<table>
<thead>
<tr>
<th></th>
<th>Walk</th>
<th>Jog</th>
<th>Run</th>
<th>Box</th>
<th>Wave</th>
<th>Clap</th>
</tr>
</thead>
<tbody>
<tr>
<td>Walk</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Jog</td>
<td>0</td>
<td>77.78</td>
<td>11.11</td>
<td>0</td>
<td>0</td>
<td>11.11</td>
</tr>
<tr>
<td>Run</td>
<td>0</td>
<td>11.11</td>
<td>88.89</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Box</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Wave</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>Clap</td>
<td>0</td>
<td>0</td>
<td>11.11</td>
<td>0</td>
<td>11.11</td>
<td>77.78</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Dive</th>
<th>Golf</th>
<th>Kick</th>
<th>Lift</th>
<th>Rid</th>
<th>Run</th>
<th>Skate</th>
<th>Swing</th>
<th>Walk</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dive</td>
<td>75</td>
<td>25</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Golf</td>
<td>0</td>
<td>75</td>
<td>25</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Kick</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Lift</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Rid</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>25</td>
<td>50</td>
<td>25</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Run</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Skate</td>
<td>0</td>
<td>0</td>
<td>25</td>
<td>0</td>
<td>0</td>
<td>25</td>
<td>50</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Swing</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>25</td>
<td>75</td>
<td>0</td>
</tr>
<tr>
<td>Walk</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
</tr>
</tbody>
</table>

We testify the approach in two cases. In the first case, the same experiment described previously was performed on the same datasets using two other representations to evaluate if the proposed spatio-temporal detector improve the original SIFT approach. For the first representation, the original 2D-SIFT detector and descriptor (Lowe, 2004) is applied to represent each frame separately. While for the second representation, the 2D-SIFT detector (Lowe, 2004) is combined with the 3D-HOG descriptor proposed by Scovanner et al. (2007). The results reported in Table 3.2 shows that the proposed detector combined with the 3D-HOG descriptor outperform the other two representations. This confirms that the presented approach can capture the interest points that have vital information in both spatial and temporal domains missing in other procedures, and can represents the events in real image sequence.

In the second case, we compared our performance with the interest points detection
approaches published recently using the same datasets. From the results presented in Table 3.3, our recognition rate for the KTH dataset exceed the state-of-the-art results reported by Nowozin et al. (2007) (87.04%), Niebles et al. (2008) (83.30%), Dollar et al. (2005) (81.17%) and Schuldt et al. (2004) (71.72%). An excellent result is also obtained for the UCF sports dataset compared to Yeffet and Wolf (2009) (79.20%), Liu et al. (2009) (74.50%) and Rodriguez et al. (2008) (66.30%).

Table 3.3: Comparative results on the KTH and UCF sports datasets

<table>
<thead>
<tr>
<th>Method</th>
<th>Detector</th>
<th>Descriptor</th>
<th>KTH</th>
<th>UCF sports</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gilbert et al. (2009)</td>
<td></td>
<td></td>
<td>94.50%</td>
<td></td>
</tr>
<tr>
<td>Mikolajczyk and Uemura</td>
<td></td>
<td></td>
<td>93.20%</td>
<td></td>
</tr>
<tr>
<td>Schindler and van Gool</td>
<td></td>
<td></td>
<td>92.70%</td>
<td></td>
</tr>
<tr>
<td>Laptev et al. (2008)</td>
<td></td>
<td></td>
<td>91.80%</td>
<td></td>
</tr>
<tr>
<td><strong>Our approach</strong></td>
<td></td>
<td></td>
<td><strong>90.74%</strong></td>
<td></td>
</tr>
<tr>
<td>Nowozin et al. (2007)</td>
<td></td>
<td></td>
<td>87.04%</td>
<td></td>
</tr>
<tr>
<td>Niebles et al. (2008)</td>
<td></td>
<td></td>
<td>83.30%</td>
<td></td>
</tr>
<tr>
<td>Dollar et al. (2005)</td>
<td></td>
<td></td>
<td>81.17%</td>
<td></td>
</tr>
<tr>
<td>Schuldt et al. (2004)</td>
<td></td>
<td></td>
<td>71.72%</td>
<td></td>
</tr>
<tr>
<td>Ke et al. (2005)</td>
<td></td>
<td></td>
<td>62.96%</td>
<td></td>
</tr>
<tr>
<td>Kovashka and Grauman</td>
<td></td>
<td></td>
<td>87.27%</td>
<td></td>
</tr>
<tr>
<td>Kläser et al. (2010)</td>
<td></td>
<td></td>
<td>86.70%</td>
<td></td>
</tr>
<tr>
<td>Wang et al. (2009)</td>
<td></td>
<td></td>
<td>85.60%</td>
<td></td>
</tr>
<tr>
<td><strong>Our approach</strong></td>
<td></td>
<td></td>
<td><strong>80.56%</strong></td>
<td></td>
</tr>
<tr>
<td>Yeffet and Wolf (2009)</td>
<td></td>
<td></td>
<td>79.20%</td>
<td></td>
</tr>
<tr>
<td>Liu et al. (2009)</td>
<td></td>
<td></td>
<td>74.50%</td>
<td></td>
</tr>
<tr>
<td>Rodriguez et al. (2008)</td>
<td></td>
<td></td>
<td>69.20%</td>
<td></td>
</tr>
</tbody>
</table>
3.2 ST-SIFT Action Recognition Framework

We outperform most of the state-of-the-art results but [Laptev et al. (2008)] for the KTH dataset and [Wang et al. (2009)] for the UCF sports dataset. This can be due to the different experimental configurations in the different approaches. Some works used the LOOCV or divided the data into two equal parts for training and testing. While the other, applied an object tracking methods before extracting the interest points. However, our approach is able to clearly distinguish between the actions with speed changes such as running and walking in both datasets.

Recently, other works with different action representation and recognition concepts have also reported better results. For example on the KTH, the best result has been obtained by [Gilbert et al. (2009)] with 94.50% accuracy, where they used a multi-level data mining technique to compute the features frequency. [Mikolajczyk and Uemura (2008)] also achieved high recognition rate 93.20% by training a vocabulary forest for both motion and appearance features. [Schindler and van Gool (2008)] reported 92.70% accuracy using a very short snippets of frames to represents each action.

On the UCF sports dataset, [Kläser et al. (2010)] boost the recognition rate to 86.70% by employing the object localization in BoF representation. Better result of 87.27% accuracy was also obtained by [Kovashka and Grauman (2010)] from learning the space-time neighbourhoods in a BoF representation.

Our focus is to extract the local features that are invariant to location, scale and orientation changes. Therefore, other applications that involve scaling and orientation changes could deliver better performance for this type of features. However, the presented approach achieves comparable results with the state-of-the-arts approaches in the field.
3. CONTRIBUTION 1: SPATIO-TEMPORAL SIFT
Chapter 4

Outlines for Contributions 2-4

In this chapter we will briefly describe the research contributions in the video processing domain. A framework for video alignment and two applications for video searching and retrieval will be introduced. Details of implementation will depend on the results that shall be obtained in the next six-months. However, the general idea about each contribution can be presented in the following sections. The plan is to spend six-months of my PhD time in each contribution, which means an eighteen months from the current stage to deliver the entire research.

4.1 Contributions 2: Sparse Coding and Manifold Representation

In this research, the framework of video stream representation will be based on three steps as illustrated in Figure 4.1. Each step can be summarised in the following points:

- The first step is to define the construction unit in the video that will be used to generate the representation. There are two layers in the video sequence, the shot that is created from group of frames and the scene that contains group of shots related semantically. Defining the unit depends on the task purpose.

- The second step is to use the extracted video unit to define the interest points that define both the appearance and the motion.

- The third step is to generate a model of mid-level features by linearly decomposing the set of features into a set of basis functions which can be modified to best
represent the video as a set of statistically independent events. Therefore, the video data can be interpreted with a few basis functions out of a large set. The Isomap is then used to produce a graph of points that can be interpreted into a trajectory in the low-dimensional space. Applying the Isomap help to find the representative global solution that has the ability to explore the underlying degree of freedom for complicated data such as the rushes video.

**Figure 4.1:** System framework for video shots representation

After that, various application can be define to use these trajectories for similarity, searching and retrieval tasks.
4.2 Contributions 3: Video Clips Similarity and Classification

The first application will be video clips similarity and classification within the video sequence. In more detail, given a video stream with multiple clips, the application will define the similar clips in one task and categorise these clips to semantically meaningful classes in another task.

As shown in Figure 4.2, the first application (on the left) introduce the framework for this application. Each clip is represented using the proposed methods and then a similarity measurement is define to determine the similarity clips in one tasks, while a classifier is employed to categorised the clips into predefined classes.

4.3 Contributions 4: Instance Search (or INS)

The second application involves searching for query clips within a video stream. 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. This will adapted to deal with video stream as testing data and video clips as query data. Figure 4.2 briefly describes the concept of this application (on the right).

In order to expand my knowledge about this task, I will participate within a team in this year (TRECVID 20120) competition. Given a set of test clips and query images of 50 different topics, the run should locate up to 1000 clips for each query. The query images will contains a person, object or place entity. Each team will have to submit a maximum of four runs as either fully automatic or interactive, which has to consume no more than 15 minutes per search. The topics for the instance search will be available at the 9th of July, while the submission deadline will the the 31st of August. The dataset will be prepared from the BBC rushes video as described in section 1.4.4.
Figure 4.2: A framework for the two applications: 1) video clips similarity and classifications within the same video file, 2) instance search task for specific query clip within the video stream.
Chapter 5

Achievements and Future Plan

This chapter summarises the first year achievements and presents tables for the work plan in the next six months and next two years.

5.1 Achievements

During the last year, I had the following activities:

1. Attending the reading group meeting every week, which involves Dr.yoshi, two PhD students (Usman and me) and a research visitor. In each meeting we discussed and reviewed one paper that was suggested by the members. The papers concerned the most recent development and covered a common interest within the group. In addition, some problems and work’s issues could be discussed during the meeting. The purpose of these meetings were to strengthen communication between the members, to exchange knowledge and experiences and to review the latest progress in members work and in the field through the published papers.

2. Preparing two presentations during the group meeting to demonstrates my approach and the results that have been reached.

5.1.1 DDP Summary

The following table summarise the first year modules (2011):
5. ACHIEVEMENTS AND FUTURE PLAN

Table 5.1: DDP module.

<table>
<thead>
<tr>
<th>Unit Code</th>
<th>Title</th>
<th>Credit Value</th>
<th>Semester/Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>EEE6081</td>
<td>Visual Information Engineering</td>
<td>10</td>
<td>Spring/2010</td>
</tr>
<tr>
<td>EEE422</td>
<td>Computational Vision</td>
<td>10</td>
<td>Autumn/2011</td>
</tr>
<tr>
<td>EEE6086</td>
<td>Video Processing &amp; Analysis</td>
<td>10</td>
<td>Autumn/2011</td>
</tr>
<tr>
<td>MAS6000</td>
<td>Mathematics in Communications</td>
<td>10</td>
<td>Autumn/2011</td>
</tr>
</tbody>
</table>

5.1.2 Workshop/ Seminar Attendance

1. The 5th Saudi International Conference 2011 (SIC05) in Coventry. It is a multi-disciplinary conference that includes paper and poster presentations, Saudi exhibition day and training workshops.

2. Participating in the DCS Research Retreat 2011 with a poster that shows my research goals.

5.1.3 Coming Workshop/ Seminar

The results from the past six-months work, as presented in Chapter 3, have been summarised in a paper titled “A Spatio-temporal SIFT and its Application to Action Recognition”. It presents the developed ST-SIFT and it is performance in human action recognition framework. The plan is to submit it in the European Conference on Computer Vision 2012 (or ECCV 2012).

5.1.4 Experiments

1. Experimenting one of the VLFeat open library applications, ”Caltech-101 classification” that used the SIFT features, K-means and SVM techniques for image database classification. It was adapted to be used for searching an image within set of videos rather than set of images.

2. Presents a space-time extension of the 2D SIFT, originally applied to the volumetric images in 2D. We build a spatio-temporal DOG pyramid to detect the local extrema. We then extract the interest points not only from the spatial plane xy but also from the xt and yt planes along the time axis. These points
approved their ability to represent the space-time events in the video and well-suited to classify the human actions. The experiment presents promising results on different datasets, (see Chapter 3 for more details).

### 5.2 Next Six-months Plan

For the next six months, an approach for synchronizing and aligning the repetitive content in video sequence will be developed. Focusing on the second contribution, the results from the first contribution will be combined with feature learning algorithm and dimensionality reduction technique to analyse the video stream. Table 5.2 demonstrates the important tasks for the next six-months.

**Table 5.2:** Work plan for the next six months

<table>
<thead>
<tr>
<th>Month</th>
<th>Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>April</td>
<td>Reading about the sparse coding</td>
</tr>
<tr>
<td>May</td>
<td>Coding for sparse coding</td>
</tr>
<tr>
<td>June</td>
<td>Reading about the Isomap</td>
</tr>
<tr>
<td>July</td>
<td>Coding for the Isomap</td>
</tr>
<tr>
<td>August</td>
<td>Experiment</td>
</tr>
<tr>
<td>September</td>
<td>Preparing for the 18th months report</td>
</tr>
</tbody>
</table>

In addition to the this work, I’ll participate in TRECVID 2012 competition for the INS task. A framework of several techniques including the SIFT approach will be implemented and submitted by the end of August.

### 5.3 Next Two-year Plan

In the last panel report, there were some issues and unclear points. At this stage of my PhD, the research outline and structures is more defined. However, there remains a number of points need more investigations and experiment such as:

- Is it possible to use three different techniques in visual representation (ST-SIFT, sparse coding and Isomap) to define different levels of features?.

- Will the generated representation be sufficient for rushes video alignment task?.
5. ACHIEVEMENTS AND FUTURE PLAN

- How to use the generated representation in different video similarity applications?

Despite these issues, potential steps showed in table 5.3 may outline the remaining tasks of my PhD.

**Table 5.3: Work plan for the next two years**

<table>
<thead>
<tr>
<th>Month</th>
<th>Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>6-months</td>
<td>Literature review</td>
</tr>
<tr>
<td>12-months</td>
<td>Time-invariant extenuation of SIFT (contribution 1)</td>
</tr>
<tr>
<td>18-months</td>
<td>Sparse Coding and Isomap experiment (contribution 2)</td>
</tr>
<tr>
<td>24-months</td>
<td>The similarity applications (contribution 3)</td>
</tr>
<tr>
<td>30-months</td>
<td>Instance Search application (or INS) (contribution 4)</td>
</tr>
<tr>
<td>36-months</td>
<td>Writing</td>
</tr>
</tbody>
</table>

The plan is to finish the representation in 18 months. This involves conducting experiments with SIFT, sparse coding and Isomap on video data. From then onwards and using the first stage output, the second stage that focuses on video similarity will start.
Chapter 6

Conclusion

Video searching and retrieval involves extracting representative interest points from the video content to reduce a volume of pixels to descriptive features. These features should lead a fast and accurate searching and retrieval processes. A starting topic for my PhD research is video searching and retrieval using rushes video as development data. The idea is to represent the video data with spatio-temporal representation that can be successfully applied to different real-world applications.

This report summarises some of the video representation, coding and dimensionality reduction techniques. In addition, it presents a spatio-temporal extension to the 2D SIFT approach and demonstrates its efficiently on the task of action recognition. The proposed detector was combined with the exciting 3D-SIFT descriptor and applied to the KTH dataset that contain scale variations and to the UCF sports dataset which is more realistic and challenging one. The results show the approach ability to describe the local features of human activity performed with different views, speed and scales. To the best of my knowledge, no work has been published for fully extending the SIFT detector for video data. Some works have been published for 3D medical images while the others were for the descriptor part.

Depending on the experiments results, different approaches and directions may be taken. However, more work and experiments will be achieved to provide a representation for the video content with the low-level features. The next six-months, which will introduce an approach to temporally align the repetitive contents in video stream, can help to determine the research outline.
6. CONCLUSION
References


REFERENCES


Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. in Philosophical Magazine 2, pages 559–572. [21]


REFERENCES


Appendix A

The ST-SIFT Algorithm
Scale-space extrema detection

Algorithm 0.1 Spatio-temporal SIFT

Given:

\[ F \rightarrow \text{Set of Features}, \quad S \rightarrow \text{interval per octave}, \quad O \rightarrow \text{number of octave} \]

\[ \sigma \rightarrow \text{temporal scale}, \quad \tau \rightarrow \text{temporal scale}, \quad t_c, \, t_e \rightarrow \text{thresholds} \]

\[ I = \{I_1, I_2, \ldots, I_N\} \rightarrow \text{input image sequence} \]

- \[ I_1 = I \]
- \[ F = \emptyset \]

- for \( i = 1, \) till \( O \) do:
  - \( I_{i,1} = I_i \)
  - for \( j = 2, \) till \( S + 3 \) do:
    * \( I_{i,j} (\cdot, \sigma, \tau) = I_i (x, y, t) \ast G (x, y, t, \sigma, \tau) \)
      where \( G (x, y, t, k^j \sigma, \tau) = \frac{1}{\sqrt{(2\pi)^3 \sigma^4 \tau^2}} \exp \left( -\frac{x^2 + y^2}{2\sigma^2} - \frac{t^2}{2\tau^2} \right) \) and \( k = 2^{j/s} \)
    * \( \text{DOG}_{I_{i,j-1}} = I_{i,j} - I_{i,j-1} \)
  - end for
  - \( pI = \text{FindExtrema} (\text{DOG}_{I_i}) \)
  - for all \( p \in pI \) do:
    1. if \( p > t_c \) do:
    2. 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
    (a) compute \( \text{Trace} (H) = D_{xx} + D_{yy} + D_{tt} \)
    (b) compute \( \text{Det} (H) = D_{xx}D_{yy}D_{tt} - 2D_{xy}D_{xt}D_{yt} - D_{xx}D_{yt}^2 - D_{yy}D_{xt}^2 - D_{tt}D_{xy}^2 \)
    (c) if \( \frac{\text{Trace}^3 (H)}{\text{Det} (H)} < \frac{(2t_e + 1)^3}{(t_c)^3} \) do:
      \( F = F \cup \text{Orientation assignment} \ (p) \)
    end if
  end if
- \( I_{i+1} = \text{Scale} (G (x, y, t, \sigma), 0.5^i) \)
- end for

return \( F \)