Selection of Video Descriptors: Generating Compact Descriptor Sets for Video Pairwise-Matching

SELECTION OF VIDEO DESCRIPTORS: GENERATING COMPACT DESCRIPTOR SETS FOR VIDEO PAIRWISE-MATCHING

BY

TING YIN, B.Sc.

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

Master of Applied Science (2017)
(Electrical & Computer Engineering)

McMaster University
Hamilton, Ontario, Canada

TITLE: Selection of Video Descriptors: Generating Compact Descriptor Sets for Video Pairwise-Matching

AUTHOR: Ting Yin
B.Sc., (Electrical Engineering and Automation)
Beijing Jiaotong University, Beijing, China

SUPERVISOR: Dr. Jun Chen

NUMBER OF PAGES: xii, 85

To my family and friends

Abstract

This thesis presents several descriptor selection schemes for video-content pairwise-matching tasks. Those proposed schemes attempt to leverage two significant properties of videos, temporal correlation and motion information.

Aiming to find an efficient and descriptive representation for a video sequence, the concept of descriptor persistency is defined. Those descriptors that satisfy this definition are called persistent descriptors. In order to exploit descriptor persistency, an encoder is proposed.

The proposed encoder consists of five main components. First, keyframe labelling is introduced to reduce complexity and ensure a reasonable size of persistent sets. After that, persistent descriptor detection is performed on group of pictures (GOP) separately. The second component is the standard SIFT descriptor extraction. The third part is to identify persistent descriptors from each GOP, called persistent descriptors extraction. In this stage, three different methods are proposed: The direct method and two approximation approaches. Persistent descriptors selection, which is the fourth stage, is carried out to control the size of the persistent set. For this stage, three selection methods are proposed. All of them attempt to utilize the motion information to select more descriptive descriptors among all the persistent descriptors in the GOP. In order to perform pairwise-matching, in this thesis, a simple but efficient

pairwise-matching method is proposed.

Experiments are carried out to evaluate the performance of the proposed schemes. The datasets used for performance evaluation are subsets from the categories that describe in [1]. Two metrics defined in [2], namely false positive rate (FPR) and true positive rate (TPR), are used for the performance evaluation.

Acknowledgements

I would like to take this opportunity to express my sincere gratitude of thanks to people who have helped me a lot to make the completion of this thesis possible. First and foremost, I would like to express my sincerest appreciation to my supervisor Dr. Jun Chen, for his kindness, great patience and wise advice. During the past two years, he has provided various of inspiring ideas and continuous encouragement to me, which will never be forgotten in my future life.

In addition, I am also very grateful to Dr. Sorina Dumitrescu and Dr. Jiankang Zhang for serving on my thesis defense committee. Their comments and suggestions are valuable to my research work.

Furthermore, I would like to appreciate all the colleagues in my lab for generously sharing their knowledge and making my life colorful. Especially, I really appreciate Muhammad and Yingchan, for the kind help they provide me. Staff members in the department are also appreciated, in particular Cheryl, who offered kind assistance in my period in McMaster University.

Lastly, I would love to dedicate to my parents with my deepest gratitude, for everything they did for me. A special thanks goes to my boyfriend, for his love and support.

Notation and abbreviations

SIFT: Scale Invariant Feature Transform

SURF: Speeded up Robust Features

CHOG: Compressed Histogram of Gradients

HOG: Histogram of Oriented Gradients

CDVS: Compact Descriptors for Video Search

CDVA: Compact Descriptors for Visual Analysis

RANSAC: Random Sample Consensus

PROSAC: Progressive Sample Consensus

LDR: Logarithmic Distance Ratio

NMF: Non-Negative Matrix Factorization

GOP: Group of Pictures

KL: Karhunen-Loeve

FPR: False Positive Rate

TPR: True Positive Rate

PKU: Peking University

KBps: Kilo-Byres Per Second

Contents

List of Figures

Chapter 1

Introduction

1.1 Background

In the modern world, especially in the new digital era, images and videos have become two of main sources of information. Motivated by the growth of online multimedia content, significant attention has been drawn to the problem of how to handle this massive amount of visual data, which, in turn, results in a variety of promising research topics in the area of visual content analysis, such as image search [3], object recognition [4] and 3D reconstruction [5].

Feature extraction is the first stage of most computer vision tasks, thus, extracting features that efficiently describe the content of a video or an image is very important yet challenging. Therefore, instead of focusing on describing high level content of image, extraction starts from local features.

The research on local feature extraction from images could be traced back to the early 1970s. The initial applications were image matching problems: In 1973, Horn made use of edge detectors to obtain edge-fragments for clean-line drawing of simple geometry [6], which in some sense proved that edges could be treated as features for describing local regions. Later on, inspired by the fact that a corner can be viewed as the intersection of two edges, in 1980, Moravec improved original edge detector and proposed a corner detector to capture corner information for stereo matching [7]. As a corner has two different edge directions, in most of the cases, a corner can be viewed as an interest point. Therefore, after Moravec's work, the concept of detecting local interest points for image matching started receiving attention. From that time, most of the advancements in feature detection are based on detecting a set of local interest points. Typical examples are Harries corner detector and its developments [8, 9, 10].

Feature detection was extended to handle more complex problems in the following years. One of the landmarks is the work of Schmid and Mohr in 1997, which addressed the image matching problem in a large image database [11]. They did not only use Harries detector to extract local interest points, but also introduced an orientation-invariant descriptor that is used to match two images. That means, if two images have almost the same object but with slightly different orientations, they can still be matched.

From that time, the definition of descriptor arises. In general, a descriptor is composed of two parts: the feature part and its associated location. In most of the work in visual analysis, those descriptors are often high-dimensional vectors.

However, these schemes were still sensitive to variations in image scale, orientation and so on, which made them unpractical for real-world applications. Not until the end of the 20th century was a new type of local features that is robust to some of those variations proposed. This local descriptor is called Scale Invariant Feature

Transform (SIFT) descriptor, and it was presented by David Lowe in 1999. This descriptor was further improved in 2004 [12]. In general, a SIFT descriptor is designed to provide a brief but distinctive representation of visual data. Compared to previous descriptors, it has been proven to be stronger in the robustness and the distinctive character experiments [13]. Then, after SIFT descriptor was proposed, a large number of different kinds of descriptors sprang up to address the descriptor extraction problem, such as Speeded up Robust Features (SURF) [14], Compressed Histogram of Gradients (CHOG) [15].

It is worth noting that these descriptors have found their way to a large class of challenging visual problems. For example, Histogram of Oriented Gradients (HOG) descriptors are widely used in human detection. As it is well known, it is not easy to detect humans in images because humans have different poses and appearances, which are difficult to capture. To solve this complicated problem, the extracted descriptors of a person must be discriminating and clean enough for further processing. As HOG descriptor can be computed on a dense grid of uniformly spaced cells, it turns out to be suitable and is widely used in human detection, such as hand gesture recognition [16] and pedestrian detection [17].

However, given an image or a video frame as input, these schemes might generate a large amount of raw descriptors, which may suffer from huge bitrate demand and computational complexity. For instance, more than one thousand SIFT descriptors can be extracted from an image of size 640×360 (230400 pixels). As a consequence, the problems of computational burden and storage requirements arise when those descriptors are used.

In order to better exploit these descriptors, the Motion Picture Expert Group

(MPEG) standardized a compact representation of image descriptors, which they called Compact Descriptors for Visual Search (CDVS) [18]. The standard was finalized and published in 2015. Later on, in an effort to extend their work to video descriptors, the group has launched another project for video analysis, that is Compact Descriptors for Video Analysis (CDVA) [1]. This standard is still under development. Both of two standards are built to solve the problem of generating distinctive, yet compact sets of descriptors. Besides, they attempt to standardize bitstream syntax for search, match and retrieval applications.

1.2 Previous Video Coding Schemes

Since CDVS standard was completed, generating compact descriptors for video sequences has been gaining much more attention in these few years. Partly motivated by the CDVA project, some research groups attempt to address the issue for video analysis, resulting in a variety of coding schemes for video descriptors. In this section, three popular approaches are briefly introduced.

The first approach is designed based on applying clustering to video frame descriptors. For example, [19] attempts to discover a small set of descriptors to represent the dominant object. In order to achieve this goal, SIFT descriptors are extracted from consecutive frames in a group of pictures (GOP) and stacked together to form a large set of descriptors. After this step, every GOP has an associated set of descriptors. Then, the Non-Negative Matrix Factorization (NMF) is proposed to perform clustering to those descriptors in each GOP and to generate a set of compact descriptors by using the centroids of clusters. The proposed clustering approaches show the merit of efficiency and conciseness compared with K-means clustering in the experiment part.

The clustering-based approach is also adopted to encode descriptors in [20]. Instead of using NMF, a two-stage approach is followed in [20] for clustering to obtain vector of locally aggregated centers. However, this kind of approach fails to make use of motion information to simplify the process of video analysis. Additionally, they treat video sequences as a set of images and process them all, which may also cause a redundancy problem.

Focusing on video descriptors compression by utilizing image-based techniques is also a popular option. For instance, in 2015, a coding scheme was proposed in [21] to address the redundancy of inter-frame problem. First of all, CDVS-based descriptors are selected from several successive frames. Then, novel concept of "reliable key-points" is defined, which represent the descriptors that exist in different frames but capture the same objects. Thus, consecutive video frames are matched to generate the reliable descriptor sets and encode these descriptors by predictive coding. Similar work is done in [22]. In some sense, motion information is used in these schemes, but they only use the limited number of reliable descriptors to describe the complex motion of video object. Since the size of descriptor set is small, those descriptors may oversimplify the motion description, which might introduce errors when it turns to the real-world applications.

Another approach is to take advantage of trajectories for motion tracking. The earlier work was done by [23]. Their work is based on the motion descriptors that were defined in MPEG-7 standardization. Motion activity and motion trajectory are together exploited to provide a comprehensive representation of the motion of each object in the video. Following this line, each object has a unique associated point along with its trajectory. Since most videos may not contain that many objects, by using this method, the processed video only includes very few points and trajectories. This high-level description will in turn lead to the reduction of motion reliability and matching accuracy. More recently, in 2016, a coding scheme was proposed in [24] that also adopts trajectories for predictive coding. In their work, first of all, several sets of CDVS-based descriptors are extracted from a video sequence. Then, a smaller set of descriptors will be obtained after feature matching stage. Also, the trajectories of those descriptors will be tracked. Finally, two different approaches of keypoint trajectories compression have been generated for different application requirements. Compared with other schemes, the approach with trajectories tracking can not only make use of temporal information but also better utilize motion correlation.

As can be seen in the schemes introduced in this section, most of works use approximately the same process for videos, which can be summarized in three stages. First, descriptors are extracted from video sequence. The popular choices are SIFT descriptor or CDVS-based descriptors. The second stage is to select descriptors which are better to meet the requirements of application. Lastly, the selected descriptor sets are compressed using a lossy coding scheme for low-rate transmission. As the first stage is usually standard, most of the novel work will take place in the last two stages.

1.3 Thesis Statement and Contribution

In this thesis, we address the problem of how to select a compact set of video descriptors that is suitable for video pairwise-matching tasks and very efficient in terms of discriminability and bitrate.

At the heart of the thesis is the novel concept of descriptor persistency. First of all, the definition of descriptor persistency is given along with its properties. An encoder designed for taking advantage of persistent descriptors is then proposed, which consists of five main components. The first is the keyframe labelling that works for reducing complexity and ensuring a reasonable size for the persistent set. Then, the second component is SIFT descriptor extraction, which is standard.

The third and fourth stages are persistent descriptor extraction and selection. In the persistent descriptors extraction, three different approaches are presented: the direct method and two approximation method. Three selection approaches, that are selection based on displacements, clusters and blocks, are proposed in the persistent descriptors selection. The last component is descriptor compression. After describing the encoder, a corresponding decoder is described along with the pairwise-matching method. Finally, experiments are carried out to evaluate the performance of these descriptor-selection schemes.

There are five chapters in the rest of this thesis, which are organized as follows: Chapter 2 provides a brief introduction of the technologies used in this thesis. Detailed descriptions of the main components of the video encoder are presented in Chapter 3. Chapter 4 briefly describes the decoder and the method of pairwise-matching. Some experimental results of the proposed video descriptor selection schemes are presented in chapter 5. Chapter 6 concludes the whole thesis and gives a glance on possible future work.

Chapter 2

Preliminaries

Prior to introducing the coding schemes in this thesis, brief descriptions of basic technologies that are used in the work are reviewed in this chapter. This chapter is divided into four parts. Since SIFT method plays a significant role in the extraction stage, Section 2.1 introduces how SIFT is applied to extract local descriptors from images or video frames. The matching scheme of SIFT called ratio test is discussed in Section 2.2. As matches obtained by ratio test may include some errors, Section 2.3 describes a method called geometric consistency check which is able to remove outliers. Lastly, since motion information should be utilized in video analysis, classical methods for motion estimation are briefly described in Section 2.4.

2.1 SIFT Method

SIFT is a popular method for capturing and depicting local descriptors. As it is mentioned in Chapter 1, the descriptors extracted by SIFT are invariant to image scaling, translation and rotation. Moreover, they are robust to lighting change, affine transformation and certain additional noises.

Among all the advantages of SIFT, scale invariability is a significant property. For the same object, no matter what the image scale is, SIFT can always extract the same descriptors from the same object. This property is obtained from scale-space theory.

As it is well known, one object in real world can be meaningful only when it is in a limited scale. For example, the concept of a cup can make sense when we observe it on the scale of few meters. However, it is insignificant to talk about a cup on a table in the scale of more than ten thousand kilometers, which implies that scale is a crucial concept in visual analysis. In 1991, Lindeberg proposed scale-space theory in a systematic way [25], indicating that one way to deal with this issue is to represent visual data in multi-scale domain. Furthermore, it has been proved that under some reasonable conditions and assumptions, Gaussian function is the unique function that can be applied to build the scale-space [26].

Following this line, in order to extract keypoints that are invariant to scaling, the first step in the SIFT method is to identify keypoints across all possible scales. We define the scale of an image $I(x, y)$ as $L(x, y, \sigma)$, where σ is the scale parameter, which is derived from the convolution of a Gaussian function $G(x, y, \sigma)$ with image $I(x, y)$:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y). \tag{2.1}$$

As shown in [27], it is efficient to compute extreme points using the result of convolving difference-of-Gaussian function with the image in scale-space. Therefore, by calculating the difference between two adjacent Gaussian functions and convolving the difference with the image, the difference-of-Gaussian function $D(x, y, \sigma)$ is obtained:

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y), \tag{2.2}$$

where two adjacent scales are separated by a constant k. After generating difference-of-Gaussian functions, the way to detect extrema points is to compare sampled points (pixels) with their eight neighbors in the current image as well as with the nine neighbors in the image of scale above and below. Then, a set of potential local keypoints are obtained.

At the second stage of SIFT, the unstable points in the potential keypoints set including the points on the boundaries and the points with low contrast are filtered out to improve robustness. In SIFT method, Taylor expansion of the scale-space function is used to reject the points with low contrast. Meanwhile, the Hessian matrix is utilized to ignore the edge points. Only the points which pass the above two tests are viewed as stable keypoints.

After the two steps mentioned above, keypoints of an image are invariant to scaling. However, in order to achieve rotation invariability, each keypoint should also be assigned a consistent orientation, which is the third step in SIFT. Within the third stage, first of all, for each keypoint p_i, the nearest scale image, $L(x, y)$, is found. Then, for all points in $L(x, y)$, the gradient magnitude, $m(x, y)$, and orientation, $\theta(x, y)$, are computed, paving the way to form an orientation histogram. This histogram is generated within a region around p_i. Each bin in the histogram represents a direction angle, and the angle with the largest value is the orientation of the keypoint.

Since each keypoint has scale, orientation and location, it is time to generate the descriptors. Each descriptor is formed using its Gaussian scale image. In Gaussian scale image, keypoint is used as a center to compute Gaussian weighting of gradients in 4×4 regions. Within each region, 8 directions are included. Therefore, each feature descriptor is a $4 \times 4 \times 8 = 128$ dimensional vector.

2.2 Matching scheme of SIFT

Once SIFT descriptors are extracted from two images, the image matching problem can be converted to descriptors matching problem. Note that there are two images, the query and the reference, and all descriptors in the query are matched with those of the reference image. As it is discussed in Section 2.1, a SIFT descriptor is a 128-element feature vector. Therefore, descriptor matching can also be treated as a vector matching problem in which Euclidean distance is used to measure the similarity between two descriptors. For a descriptor in the query image, the nearest neighbor in the reference image can be defined as the descriptor with smallest Euclidean distance among all descriptors in reference image.

Although every descriptor in the query image can always find a nearest neighbor from the reference image in the sense of Euclidean distance, we cannot declare they are all matches. Thus, it is important to find a way to distinguish such descriptors that do not have any matches in the reference image. In [28], Lowe generated a matching scheme called ratio test. At the heart of ratio test is the idea that correct matches need to be significantly better than other features. It means that the nearest neighbor should be significantly closer than any other possible descriptor. Therefore, we can take the second-closest neighbor as an estimate of false match to measure whether the nearest match is good or not.

The test can be described as follows:

1. All descriptors in both query image and reference image are extracted.

2. Take a descriptor d_i from the query. The Euclidean distances between d_i and all descriptors in the reference image are calculated and only two descriptors with the smallest and the second-smallest distances are maintained. These distances are referred to as $dist_1$ and $dist_2$, respectively.

3. The ratio r of the smallest distance to the second-smallest distance is obtained by:
$$r = \frac{dist_1}{dist_2}. \tag{2.3}$$

4. If $r < 0.8$, d_i is a match to the descriptor with the smallest Euclidean distance in the reference image. Otherwise, d_i has no match in the reference image.

5. Repeat step 2 until all query descriptors are processed.

What needs to be emphasized is that a descriptor in the reference may have several matches in the query, that is, two or more descriptors in the query are matched to the same descriptor in the reference image because all these pairs pass the ratio test. To deal with this issue, a matching score is assigned to each matching descriptor pair, which is computed as:
$$\beta = \cos(\frac{\pi r}{2}), \tag{2.4}$$
where r represents the distance ratio obtained from (2.3). This matching score is used for removing duplicate matches. To be more specific, when multiple descriptors in query image are matched to the same descriptor in the reference, a unique match that has the largest score will be kept and declared as the best match and the other matches will be discarded.

Therefore, each descriptor matching pair will have an associated matching score.

By summing up all the matching scores of descriptors that are declared matches, the total matching score for the two images is obtained.

2.3 Geometric Consistency Check

As discussed in Section 2.2, matching descriptor pairs are formed by finding the closest neighbor using Euclidean distance. However, even for the descriptor pairs that pass ratio test, there still exists some wrong matches. For example, as it is shows in Fig. 2.1a, it is clear that there are three incorrect matches although they pass the ratio test.

In order to separate these wrong matches from correct ones, the concept of geometric consistency checking arises. Several methods are used to solve this issue, like Random Sample Consensus (RANSAC) [29] and Progressive Sample Consensus (PROSAC) [30]. These methods are designed to verify the consistency of a geometric model and obtain the properties of correct matching descriptors. Since incorrect matches perform differently from correct ones, removing bad matches becomes possible. Note that a match is called an inlier if the descriptor is correctly matched in ratio test. Otherwise, this match will be called outlier.

As shown in [31], CDVS adopted the histogram of logarithmic distance ratios (LDR) for geometric consistency check, which consists of two stages. In the first stage, the method aims to determine whether this two images have the same scenes (match) by using a hypothesis test. The details are introduced in [32]. In their work, by making use of LDR, a histogram of LDR values is built. Furthermore, a probability density function is generated for incorrect matches. These two functions together are used in a goodness-of-fit test to check whether these two images are indeed a match or not. Compared to other schemes, [32] differs in focusing on generating a statistic model for incorrect matches. When the pairs of images passes the first stage, inliers will be estimated in second stage. The steps for finding inliers can be briefly described as follows:

1. After ratio test, N pairs of matched keypoints are obtained, which are denoted by

   $$\{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n), \cdots, (x_N, y_N)\},$$

   where x_n represents the two coordinates of the location of n^{th} descriptor in the query image, and y_n is the coordinates of its matching descriptor location in the reference image.

2. The LDR values for these matches are calculated. More precisely, LDR statistic function is computed as:

   $$Ldr_{ij} = \ln(\frac{\|x_i - x_j\|}{\|y_i - y_j\|}), \quad i, j \in \{1, 2, \cdots, N\}. \tag{2.5}$$

   What needs to be emphasized is that $x_i \neq x_j$ and $y_i \neq y_j$ must be satisfied since descriptors must be distinct. Thus, a large set of LDR values can be obtained after this stage.

3. A histogram with K bins is generated by LDR values, denoted by $h(k)$, where $k \in \{1, 2, \cdots, K\}$. This histogram will be compared to the model function [32]. Furthermore, the model function (continuous function) is discretized to obtain a histogram with the same length as $h(k)$. Therefore, a set of model probabilities $p_1, p_2, ..., p_k...$ will be obtained, which is denoted by $p(k)$ and $k \in \{1, 2, \cdots, K\}$.

4. The two histograms, $p(k)$ and $h(k)$, are together passed to a goodness-of-fit test, and $i \in \{1, 2, \cdots, N\}$. If the result exceeds a threshold given by chi-square function, move to the next step, which is the inliers estimation stage. Otherwise, these two image will be declared nonmatching.

5. The inliers estimation is performed using an eigenvalue problem. First of all, the factor β is computed as

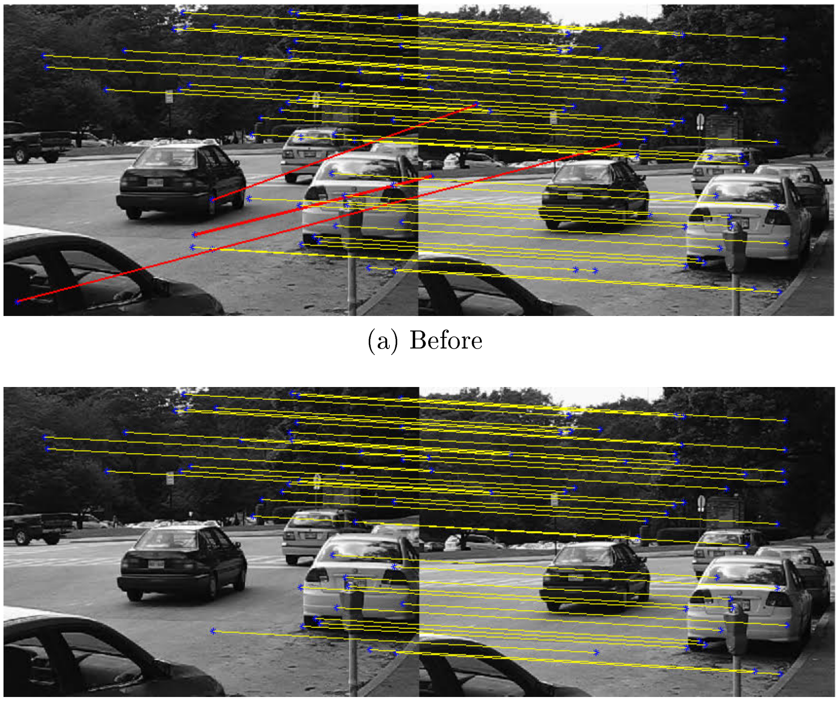
$$\beta = \frac{\sum\limits_{k=1}^{K} h(k)p(k)}{\sum\limits_{k=1}^{K} (p(k))^2}, \tag{2.6}$$

then the outliers normal function is created as follows:

$$d(k) = h(k) - \beta p(k), \quad k \in \{1, 2, \cdots, K\}. \tag{2.7}$$

6. Compute inlier evidence metrix D. Each element in D is denoted by D_{ij}. If $i \neq j$, D_{ij} equals the corresponding $d(k)$ of Ldr_{ij}, which is denoted as $Ldr_{ij} \rightarrow d(k)$, and $i, j \in \{1, 2, \cdots, N\}$. Otherwise, if $i = j$, D_{ij} equals to zero. Mathematically,

$$D_{ij} = \begin{cases} Ldr_{ij} \rightarrow d(k) & i \neq j \\ 0 & i = j \end{cases}. \tag{2.8}$$



(a) Before

(b) After

Figure 2.1: An example of the effect of geometric consistency check.

7. The dominant eigenvector of D is computed as well as its corresponding eigenvalue. Then, the number of inliers $\hat{m}$ is estimated by

$$\hat{m} = 1 + \frac{\mu}{\max\limits_{k=1,2,\cdots,K} d(k)}. \tag{2.9}$$

8. The estimated inliers indices (descriptor indices) correspond to the indices of $\hat{m}$ largest element in the obtained eigenvector.

When geometric consistency check is applied to the matches, majority of incorrect matches will be filtered out. Fig. 2.1b shows the effect of geometric consistency check.

2.4 Motion Estimation Methods

One of the main difference between videos and images is temporal information and motion correlation. Therefore, compared to image analysis, more attention should be paid to utilizing temporal information and motion correlation in video analysis. Two typical motion estimation approaches are introduced in this section.

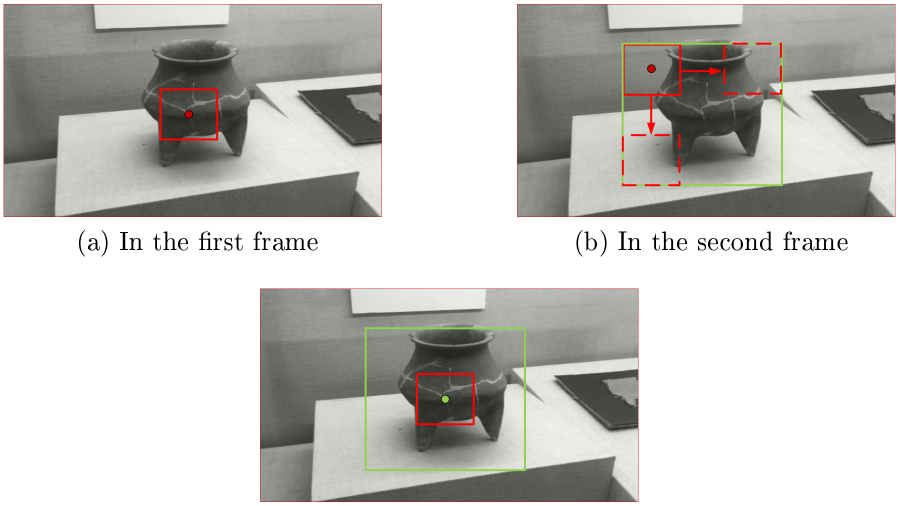
2.4.1 Optical Flow

The concept of optical flow was proposed by Gibson in 1950 [33]. In general, optical flow is designed to estimate the apparent object motion between two video frames. By calculating the motion vector of interest points and regions in different times, an estimation model is generated. In computer vision, optical flow is performed on pixel-level in digital images.

There are a variety of schemes for better estimation [34, 35]. In this section, the work of [36] is introduced.

In their way to estimate motion of pixels, a signal transformation method called polynomial expansion is adopted to estimate the neighborhood of each pixel in two consecutive frames. To be more specific, for each pixel, a quadratic polynomial is generated to approximate its neighbors and pixel intensities, and the coefficients of this polynomial are estimated by weighted least squares of the neighborhood pixels. Then, the displacement of each pixel will be estimated over its approximated neighborhood. Thus, dense optical flow field will be computed by the displacement of pixels.

Compared to the previous optical flow algorithm, the discussed approach ignores the original limitation of other schemes, that is, the motion field must be consistent.


(a) In the first frame

(b) In the second frame

(c) In the second frame

Figure 2.2: The schematic diagram of block-motion estimation.

Therefore, this approach is more robust to deal with real-world videossuch as high frequency vibrations videos.

2.4.2 Block-Motion Estimation

In digital image processing, block-motion estimation is another popular approach for motion estimation. An example of how this algorithms works for estimation is showed in Fig. 2.2. Define the point that needs to be estimated as p_i. The location of p_{1i} is displayed as a red point in Fig. 2.2a. There are two input frames, the goal is to estimate the location of p_i in the second frame.

The process of block-motion estimation works as follows:

1. Use the location of p_i to generate the patch of p_i in the first frame. As described in Fig. 2.2a, the location of p_1 is treated as the center and a square with length l is formed, which generates patch A. The obtained patch is used for further search.

2. Generate a search window in the second frame. In Fig. 2.2b, the bigger green square with length n, $n > l$, is the search window in the second frame.

3. Search patches exhaustively in the search window and find out the most similar patch B of patch A. Hence, patch B will be considered as the match of patch A. Here, the sum of absolute difference (SAD) is used to measure the similarity between two patches [37].

4. The center of patch B, p_1, is considered as the estimated location of p_{1-est}. In Fig. 2.2c, the green point is the predicted location of p_1.

Since optical flow is too sensitive to illumination changes, in this thesis, the block-motion estimation is used for motion estimation.

Chapter 3

Encoding of Video Descriptors

The most significant difference between still images and video sequences is that videos contain temporal information. In order to fully make use of that, the novel concept of descriptor persistency is proposed as the core concept of the descriptor-selection schemes in this thesis.

The encoder tries to exploit the concept of descriptor persistency to find an efficient and compact representation to all the descriptors of a video sequence. Fig. 3.1 shows the block diagram of the encoder, which consists of five main components: keyframe labelling, SIFT descriptors extraction, persistent descriptors extraction, selection and compression. The designed schemes are all based on SIFT descriptors due to its widely use in local descriptors extraction. However, it is worth noting that the designed schemes could also be applied to other kinds of descriptors, such as SURF. For the compression, the quantization method proposed in [38] is used. The rest three components are detailed in the following few sections.

This chapter is organized as follows. In the first section, the concept of descriptor persistency is given along with its associated properties. In Section 3.2, the

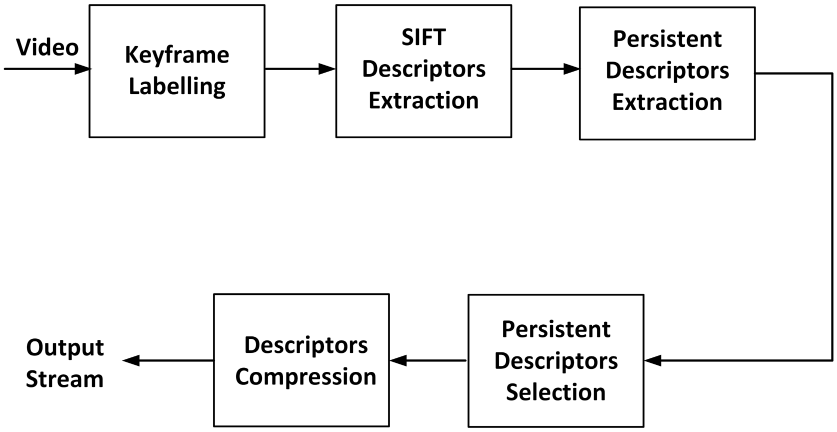

Video
Keyframe Labelling
SIFT Descriptors Extraction
Persistent Descriptors Extraction
Persistent Descriptors Selection
Descriptors Compression
Output Stream

Figure 3.1: The block diagram of the proposed encoder.

keyframe labelling process is explained. Then, different approaches to extract persistent descriptor sets are discussed in Section 3.3. The selection methods within these extracted descriptor sets are explained in details in Section 3.4.

3.1 Persistent Descriptors

As discussed in Section 2.2, after matching process, two descriptors that are extracted from two different images can be considered nearly the same when the Euclidean distance between these two features is very small. If two descriptors are declared a match, with high probability these two are extracted from the same object and capture the same characteristic. Motivated by this fact, we start considering the correlations within consecutive video frames.

Two consecutive frames in a video are likely to have similar scenes. Thus, if the descriptors from two consecutive frames are matched, with high probability, a set of matching descriptor pairs between these two frames can be found. Those descriptors tend to capture common local regions in these two frames. Besides, the number of matches depends on the correlation between the two frames. The more similar the scenes in these two frames are, the more matching pairs are found.

Extending the size of consecutive frames to multiple, the terminology descriptor persistency is proposed and defined as:

Definition 1. *A descriptor d that is extracted from a frame, called the base frame f_{base}, is said to be k-persistent if it can find matches in all the K following frames after f_{base}. These matches must have very small Euclidean distance to d and should describe the same local region that d describes in f_{base}.*

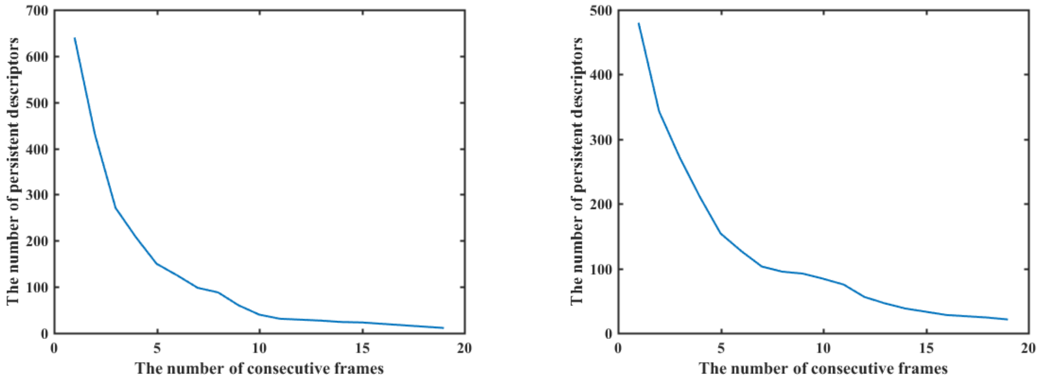
If Definition 1 is applied to all the descriptors in the base frame f_{base}, a subset of descriptors that are K-persistent can be obtained, which are called persistent descriptors. In this subset, each descriptor can find a match in all the following K successive frames. Mathematically, if the group of all descriptors detected from base frame f_{base} is defined as descriptor set S, and the subset that satisfies the definition of K-persistent is S_{per}, then,

$$S_{per} \subseteq S. \tag{3.1}$$

Based on the above definition of the persistent descriptors, several experiments on different-content videos have been performed and the following two interesting properties have been found:

1. The size of a persistent set decreases dramatically when the number of processed consecutive video frames increases.

   One inherent property of videos is that objects are moving almost all the times. During the changes of objects and scenes, some descriptors disappear and some new ones appear, which results in a subset of the descriptors in the base frame



(a) The first video case

(b) The second video case

Figure 3.2: The size of persistent set drops dramatically as the number of processed consecutive frames increases.

that does not have matches in all consecutive frames. Therefore, it makes sense that as the number of consecutive frames increases, the size of persistent set tends to drop rapidly. Two examples are listed in Fig. 3.2, showing that the number of persistent descriptors experiences dramatic changes as the number of the processed consecutive frames increases. To be more specific, in Fig. 3.2b, when the base frame is matched with its next adjacent frame, there are 640 descriptors find matches. The number decreases to 11 as the number of processed frames reaches to 20.

2. The locations of persistent descriptors comply to the motion field in the video sequence.

   Persistent descriptors are extracted from different regions in the frame in which some are stationary and some are moving. Experiments show that the locations of persistent descriptors comply with the motion field of the K-frames video segment. In other words, for those descriptors that can find matches in the

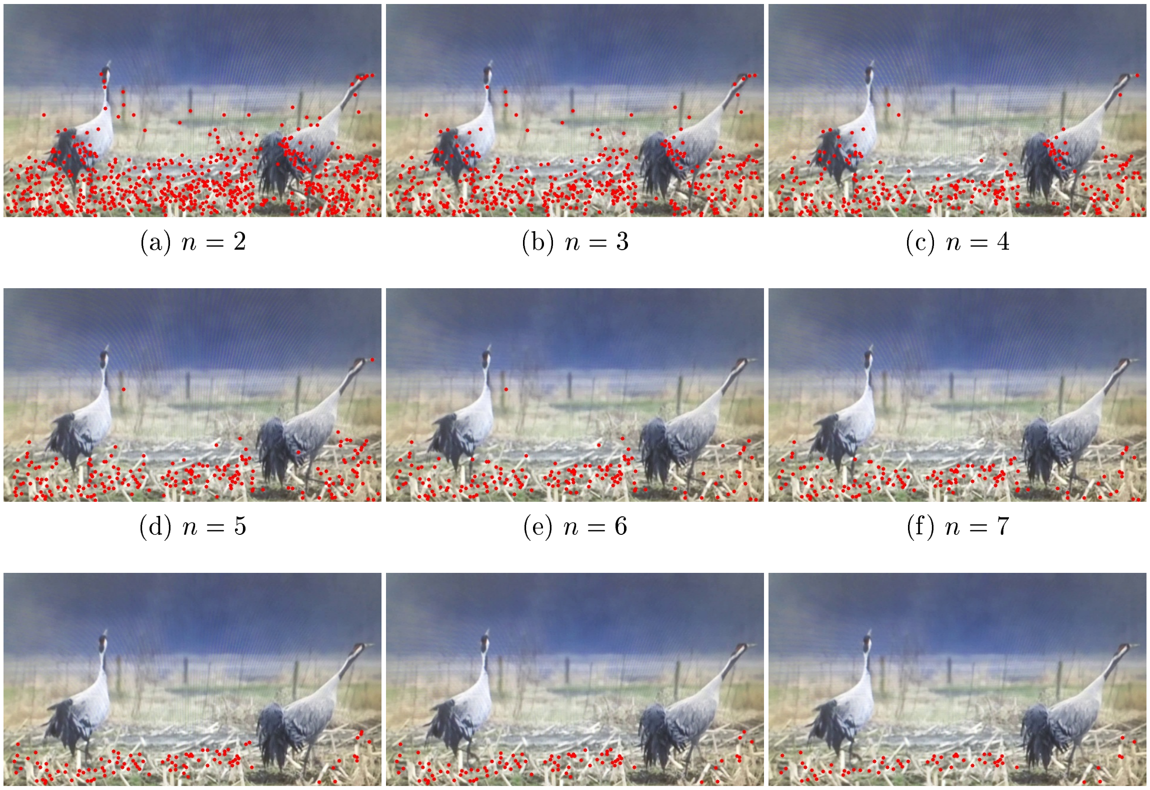


Figure 3.3: The locations of persistent descriptors as the number of processed consecutive frames increases.

following K successive frames, they are detected in the same region as their counterparts in the base frame and describe the same characteristics of the objects. Examples are displayed in Fig. 3.3, where n represents the number of successively processed frames. Compared to the base frame, locations of the remaining persistent descriptors stay the same.

Since persistent descriptors exist in several successive video frames and they capture the characteristics of the same interest regions, these descriptors are potentially describing the visually salient objects that appear in all the K frames. Those objects always convey the most important video information, thereby persistent sets

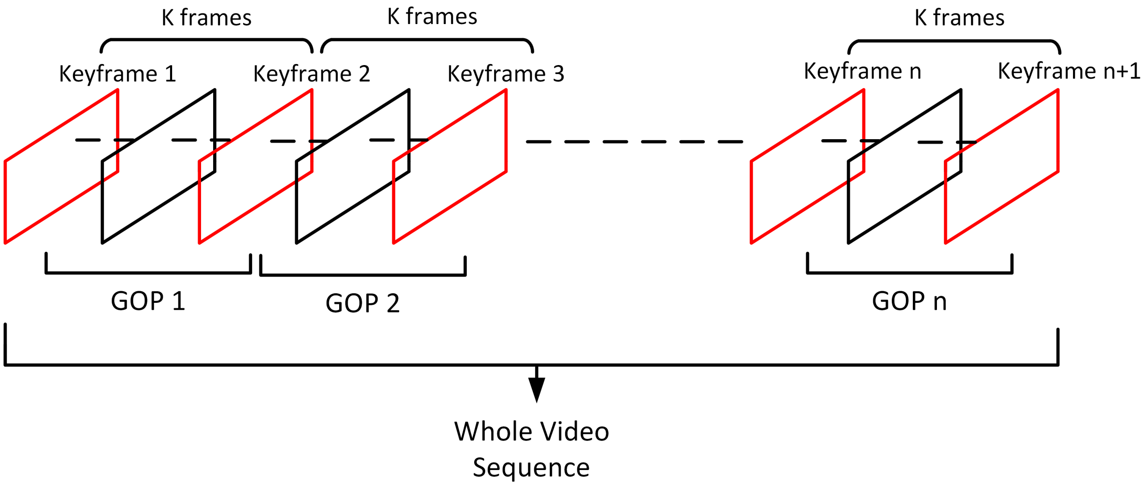


Figure 3.4: The illustration of keyframe labelling and GOP assignment.

can roughly preserve that information. From the observation of the behaviors of persistent descriptors, a conclusion can be drawn: The set of persistent descriptors might be a good compact representation of the sequence of K consecutive frames.

3.2 Keyframe Labelling

As discussed in Section 3.1, persistent descriptors are helpful for generating a compact description for a video sequence. In order to leverage the concept of descriptor persistency for further processing, persistent descriptor sets should be extracted first.

Persistent descriptors identification and extraction are kind of challenging in practice. The challenge comes from two aspects. If we detect persistent descriptors by extracting and matching all descriptors from consecutive frames, computational complexity is high. On the other hand, if the processed consecutive frames are too long (K is too large), the size of the persistent set is very small. Fig. 3.2 demonstrates an example. Although the sets of persistent descriptors are good to some extent and represent some salient objects, due to the small sizes, these descriptors cannot perform well when they are used for video pairwise-matching. Thus, in prior to extracting persistent descriptors, it's better to restrict the number of the processed successive frames to ensure a reasonable size of the persistent set. One way to overcome these two problems is to choose pairs of keyframes each of which encloses a GOP. Fig. 3.4 illustrates the keyframe labelling and GOP assignment process. The brief process can be described as follows: first of all, keyframes are chosen and labeled. The first frame of the video sequence is labeled as "keyframe1". Then, the rest keyframes are labeled with the principle that every two adjacent keyframes are with the same distance of K frames. Lastly, GOPs are formed by the set of frames sandwiched by every two keyframes. Note that adjacent GOPs share one keyframe. For instance, in Fig. 3.4, both GOP1 and GOP2 contain keyframe 2.

After applying keyframe labelling, persistent descriptors detection and approximation are done in every GOP separately. Furthermore, the size of persistent descriptors can be reasonable if K is chosen appropriately.

3.3 Persistent Descriptors Extraction

After keyframe labelling process, the video sequence gets decomposed into a sequence of GOPs. Extraction of persistent descriptors is performed in every GOP individually. Within each GOP, a set of persistent descriptors is extracted to represent the information the for whole GOP.

To be more specific, in this section, three different persistent descriptors extraction approaches are proposed. In the beginning, the direct method is introduced, which is very accurate but computationally expensive. In the rest subsections, two less complex approximation methods are presented to take advantage of the two keyframes in each GOP and approximate the persistent set of that GOP.

3.3.1 The Direct Method

From Definition 1 in Section 3.1, it is straightforward to come up with an idea to find persistent descriptors by extracting and matching SIFT descriptors across all frames in a GOP. Assume that there are K frames in a GOP, the detailed procedures for finding the persistent set of the GOP are given by:

1. Extract SIFT descriptors in all frames in the GOP. Each frame produces a SIFT descriptor set, denoted by S_{f_i}, where i is the index of the frame in the GOP, $i \in \{1, 2, \cdots, K\}$.

2. Match the descriptors of the first frame in the GOP with those of the second using the ratio test and geometric consistency check, that is, matching S_{f_1} with S_{f_2}. Thereby, pairs of unique matches between S_{f_1} with S_{f_2} can be found. Let those pairs be denoted by:

$$S_{match} = \{(d_1^{f_1}, d_1^{f_2})_1, (d_2^{f_1}, d_2^{f_2})_2, ..., (d_{N_1}^{f_1}, d_{N_1}^{f_2})_{N1}\},$$

   where N_1 is the number of matching pairs, $d_t^{f_1} \in S_{f_1}$ and $d_j^{f_2} \in S_{f_2}$, $t, j \in \{1, 2, \cdots, N_1\}$. All $d_j^{f_2}$ can form a descriptor subset $D_{match}^{f_2}$, which represents descriptors from S_{f_2} that can find matches in S_{f_1}.

3. Match $D_{match}^{f_2}$ with S_{f_3} by applying ratio test and geometric consistency check. A smaller subset of matches, denoted $D_{match}^{f_3}$, can be formed by the descriptors

from S_{f_3} that can find matches in $D^{f_2}_{match}$.

4. Repeat step 3 until $D^{f_K}_{match}$ is obtained.

$D^{f_K}_{match}$ is thereby the persistent descriptor set for this GOP. By applying the procedures described above to the whole video, several persistent sets can be generated. The number of these sets equals to the number of GOPs.

Actually, the direct method to get the persistent sets fully obeys Definition 1, indicating that it can be viewed as a very accurate method for finding persistent descriptors. However, we cannot declare that the sets obtained by the direct method are the exact persistent descriptors since the matching process could still produce some errors.

Despite its accuracy, it is really computationally expensive. Complexities comes from two main sources. On one hand, extracting SIFT descriptors from all video frames will result in a quite redundant representation of video contents in terms of rate. Removing the redundancy resulting from processing every frame in the GOP is complex. On the other hand, searching across all frames for finding matches is time-consuming and is expected to be inefficient.

Consequently, two approaches are proposed to overcome the complexity issue. Instead of processing all the frames in each GOP, these proposed approximation methods attempt to extract descriptors only from keyframes. Followed by this line of work, they push much of the computational complexity down. However, since the direct method is the most accurate method, it can be considered as the baseline.

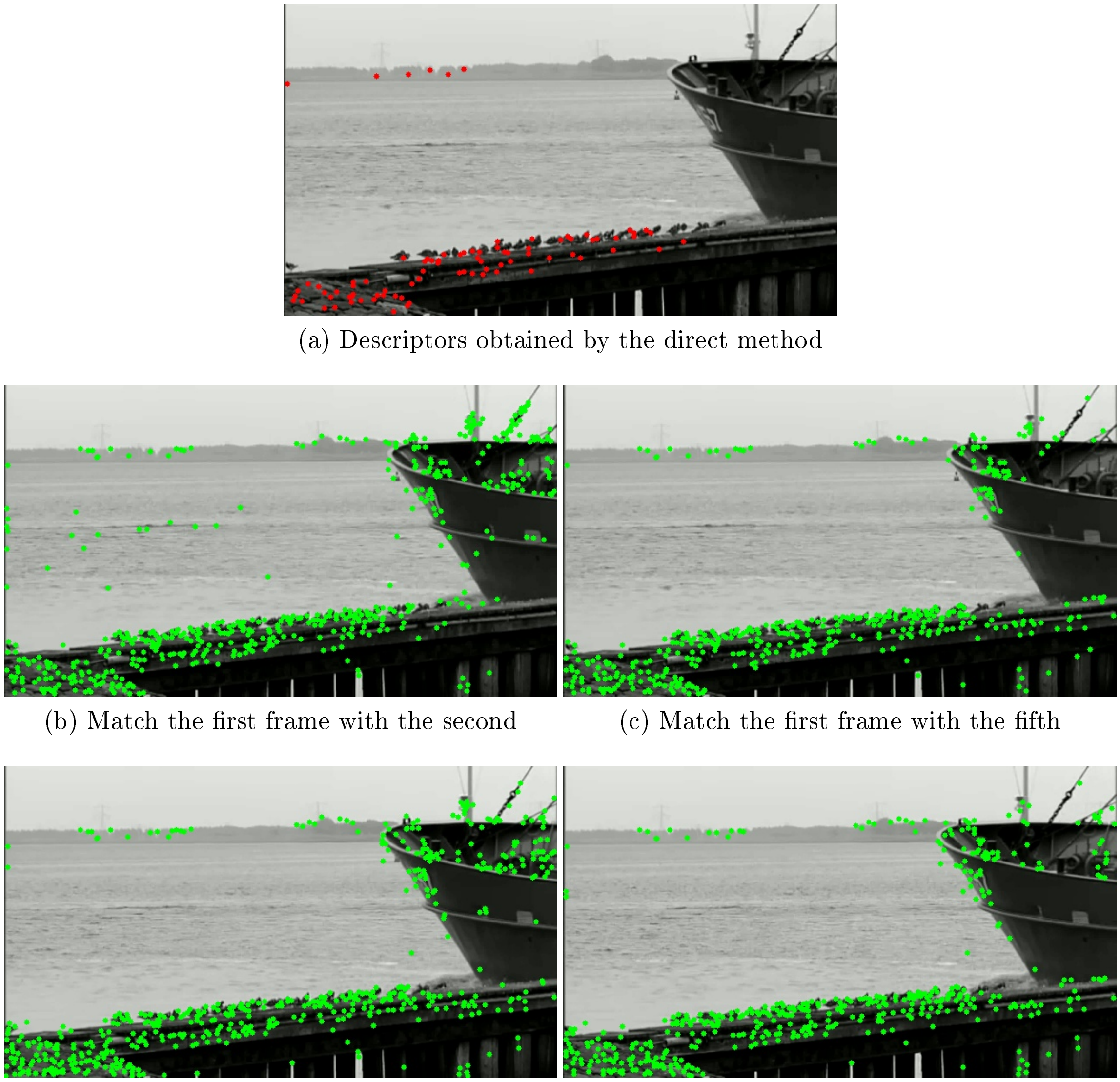
3.3.2 The First Approximation Approach

In order to reduce the complexity, an idea arises that instead of extracting descriptors across all the frames in the GOP, we only process some of the intermediate frames.

In each GOP, with high probability, frames have almost the same content and characteristics. These high frame correlations suggest that only two frames in the GOP might produce the majority of persistent descriptors in this GOP.

This statement is derived from experimental observations. The experiments are implemented by matching two frames in a GOP. To be more specific, assume that the GOP size is defined as K and K is fixed. The experiments start by randomly choosing two frames f_i and f_j, where i and j refer to the indices of frames in the GOP, $i, j \in \{1, 2, \cdots, K\}$ and $i < j$. Extracting SIFT descriptors from both frames produces two descriptor sets S_{f_i} and S_{f_j}. Then, by matching these two descriptor sets, a subset $S_{i,j}$ is generated, which consist of the descriptors in f_i that have matches in f_j. By comparing $S_{i,j}$ with the persistent descriptors that are obtained by the direct approach discussed in Section 3.3.1, the number of descriptors in $S_{i,j}$ that can find matches is obtained. Two examples are illustrated in Fig. 3.5 and Fig. 3.6. In each example, four experiments are displayed. These two video sequences are with different contents and the GOP size is defined as $K = 11$.

On the top of Fig. 3.5 is the persistent descriptors obtained using the direct method. The locations of descriptors are displayed in red points. Also, there are four different results for the same video GOP, that are, the subset $S_{1,2}$ from matching the first frame with the second, the subset $S_{1,5}$ from matching the first frame with the fifth, the subset $S_{2,7}$ from matching the third frame with the seventh and the subset $S_{5,10}$ from matching the fifth frame with the tenth. The corresponding sizes



(a) Descriptors obtained by the direct method

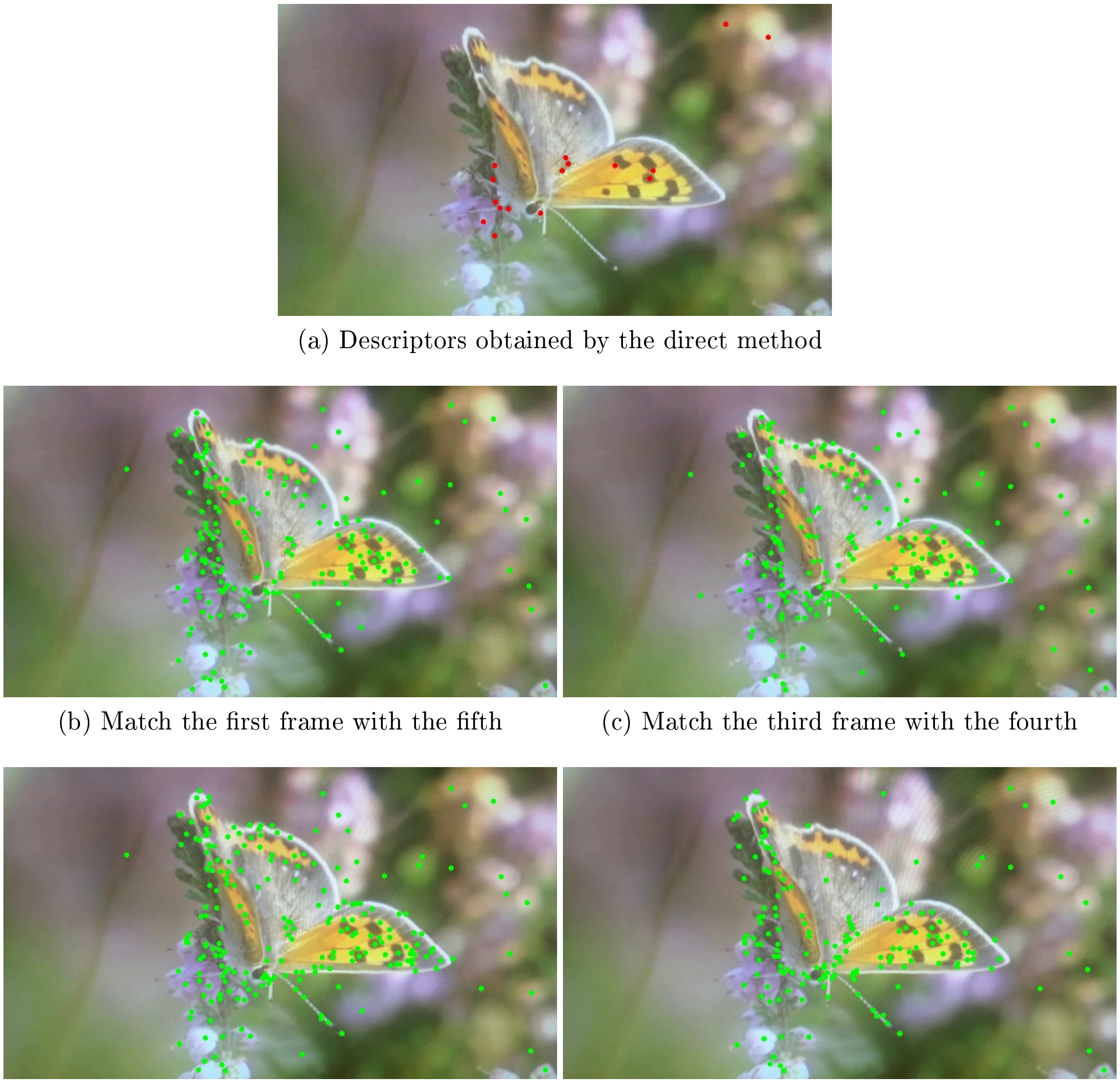
(b) Match the first frame with the second

(c) Match the first frame with the fifth

(d) Match the third frame with the seventh

(e) Match the fifth frame with the tenth

Figure 3.5: Examples 1 that indicating two frames can roughly represent the whole GOP.



(a) Descriptors obtained by the direct method

(b) Match the first frame with the fifth

(c) Match the third frame with the fourth

(d) Match the sixth frame with the eighth

(e) Match the seventh frame with the eleventh

Figure 3.6: Examples 2 that indicating two frames can roughly represent the whole GOP.

of each subset are 480, 401, 511 and 448, respectively. The green points are the descriptor locations in each subset. As can be seen from these four figures, descriptor sets obtained from different cases locate at almost the same locations. Although they are not exactly the same, they roughly capture the same salient objects, for instance, the ship and wooden dock in this video sequence. Most importantly, compared with Fig. 3.5a, these subsets do not only include the descriptors in the persistent sets, but also capture more video-content information, which are more valuable in further process. The similar situation happens in Fig. 3.6. These four different matching cases almost cover the same characteristics in the frames, and most descriptors focus on the butterfly. Besides, most of the persistent descriptors derived from the direct method are contained in all the four different cases.

From these two examples, it can be concluded that extracting descriptors from two frames in a GOP could produce the majority of the persistent descriptors. Therefore, only matching two frames in a GOP to get a persistent set is a good alternative to the direct method. Moreover, treating any two intermediate frames in the GOP as input can generate roughly the same descriptors.

In order to make use of keyframes and standardize the procedures, the approximation method is developed on two keyframes. Another more important reason for choosing two keyframes is that these two are the first and the last frame in the GOP. If a descriptor is indeed a persistent descriptor, it should appear in the first frame and the last frame for sure. Therefore, matching these two frames can guarantee the persistent descriptors are included in the obtained matching descriptor set.

The detailed process of approximating a persistent set from a GOP is shown as:

1. Extract two sets of SIFT descriptors, S_{kf1} and S_{kf2} from the first keyframe f_{kf1}

and the second keyframes f_{kf2}, respectively. These two sets can be of different sizes.

2. Match S_{kf1} and S_{kf2} by the ratio test. Then check the outliers by geometric consistency check. Thus, unique pairs of matches between S_{kf1} and S_{kf2} are obtained, which are denoted by:

$$S_{match} = \{(d_1^{kf1}, d_1^{kf2})_1, (d_2^{kf1}, d_2^{kf2})_2, ..., (d_{N_1}^{kf1}, d_{N_1}^{kf2})_{N_1}\},$$

   where $d_i^{kf1} \in S_{kf1}$ and $d_j^{kf2} \in S_{kf2}$, $i, j \in \{1, 2, \cdots, N_1\}$. Note that N_1 must be smaller or equal to the size of S_{kf1} and S_{kf2}. All d_i^{kf1} can form a descriptor subset D_{match}^{kf1} whereas all d_j^{kf2} can form a descriptor subset D_{match}^{kf2}, where $i, j \in \{1, 2, \cdots, N_1\}$.

3. Form the approximate persistent set $S_{per-approx-1}$ in the GOP by using the descriptors in the first keyframe that can find matches in the second keyframe. That is, $S_{per-approx-1}$ is composed of the subset D_{match}^{kf1}.

In this approximation method, there are three advantages. First, the complexity of identifying persistent descriptors is reduced by only processing two frames in the GOP, not all of them. In some sense, this method exploits the temporal information of a video sequence, which is the second advantage of this approach. Furthermore, as seen from Fig. 3.5a and Fig. 3.6a, the size of persistent sets obtained by the direct method is very small. For example, in Fig. 3.6a, the number of descriptors obtained by the direct method is only 16. Although these descriptors are indeed appearing in all frames in the GOP, the limited number of descriptors is expected to reduce the ability to describe the objects in the GOP. If the solution is only performed based on two keyframes, more persistent descriptors are obtained, which contain more information. It is the third advantage of the proposed solution.

Next, two different approaches are presented to measure the accuracy of the proposed approximation method.

The first one is matching descriptors obtained by the approximation method with those obtained by the direct method. As mentioned in the last paragraph of Section 3.3.1, the direct method to extract persistent sets can be considered as the baseline of persistent descriptors due to its high accuracy to satisfy the definition of descriptor persistency. The detailed process in each can be described as:

1. Extract the persistent set in the GOP using the direct method. Define the obtained descriptor set as S_{dir}.
2. Approximate the persistent descriptors in the GOP using the approximation method. The formed descriptor set is denoted by $S_{per-approx-1}$.
3. Match S_{dir} and $S_{per-approx-1}$ by ratio test. Define the subset of descriptors from S_{dir} that can find matches in $S_{per-approx-1}$ as $S_{match-1}$.
4. The match percentage is then given by $\frac{N}{M} \times 100\%$, where M and N are the sizes of S_{dir} and S_{match}, respectively.

By applying the above process to a video sequence, the results can be obtained. Here are three examples shown in Table 3.1, Table 3.2 and Table 3.3, respectively. The difference between these three cases is the size of the GOP. Table 3.1 is with GOP size of 5, that is, $K = 5$, and the other two tables are of $K = 10$ and $K = 15$, respectively. The same 60-frames video sequence is processed in these three different cases. Therefore, twelve GOPs are generated for $K = 5$ and six GOPs are generated

Table 3.1: The comparison of $S_{per-approx-1}$, S_{dir}, $S_{match-1}$ and match percentage when $K = 5$

GOP index	size of $S_{per-approx-1}$	size of S_{dir}	size of $S_{match-1}$	match percentage
GOP No.1	401	210	198	94.28%
GOP No.2	439	199	196	98.49%
GOP No.3	486	147	143	97.28%
GOP No.4	501	165	165	100.00%
GOP No.5	508	259	249	96.14%
GOP No.6	549	225	221	98.22%

Table 3.2: The comparison of $S_{per-approx-1}$, S_{dir}, $S_{match-1}$ and match percentage when $K = 10$

GOP index	size of $S_{per-approx-1}$	size of S_{dir}	size of $S_{match-1}$	match percentage
GOP No.1	382	92	92	100.00%
GOP No.2	398	67	67	100.00%
GOP No.3	433	108	107	99.07%
GOP No.4	468	87	87	100.00%
GOP No.5	436	100	99	99.00%
GOP No.6	446	65	65	100.00%

for $K = 10$. Here, the first six GOPs when $K = 5$ are shown in 3.1 from GOP No.1 to GOP No.6. For $K = 10$, all the GOPs from No.1 to No.6 are shown in Table 3.2. For $K = 15$, there are 4 GOPs, that are from GOP No.1 to GOP No.4.

From these three tables, there are two interesting observations:

1. The size of $S_{per-approx-1}$ is much larger than size of S_{dir}. This observation can also be seen from Fig. 3.5 and Fig. 3.6. As discussed in Section 3.3.1, persistent descriptors obtained by direct method have a stricter requirement, that is, they must exist in all the frames in the GOP. Therefore, as we expect, compared to the direct method, matching two keyframes produces much more descriptors.

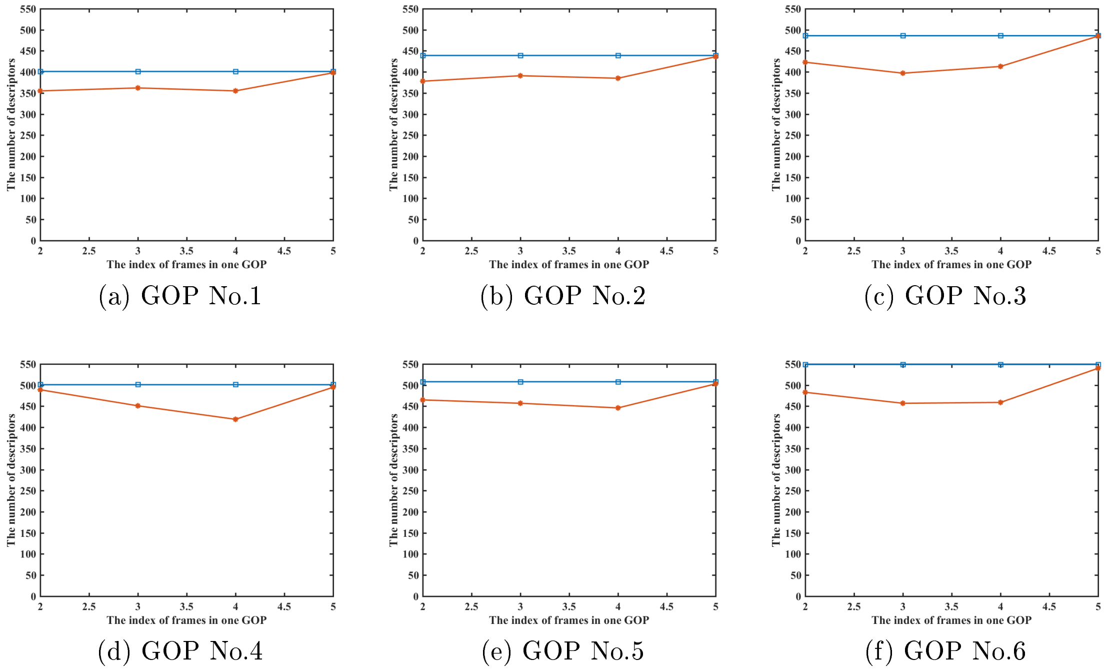
Table 3.3: The comparison of $S_{per-approx-1}$, S_{dir}, $S_{match-1}$ and match percentage when $K = 15$

GOP index	size of $S_{per-approx-1}$	size of S_{dir}	size of $S_{match-1}$	match percentage
GOP No.1	336	38	38	100.00%
GOP No.2	373	19	19	100.00%
GOP No.3	401	40	40	100.00%
GOP No.4	426	41	40	97.56%

2. In all three cases, the sizes of $S_{match-1}$ are closer to the size of S_{dir}. Furthermore, the associated match percentages are all above 94%. As it can be seen from these three tables, the highest match percentage is 100%, which happens a lot. Even for the lowest match percentage, it exceeds 94%. It can be claimed that the approximation method can extract most of the persistent descriptors obtained by the direct method. Furthermore, $S_{per-approx-1}$ also includes other descriptors that describe other features of the frames, providing more information for the overall video content.

Therefore, by these two observations, we can declare that the the first approximation method is able to capture most of the descriptors obtained by the direct method, which is our baseline.

Another way to measure the accuracy of the approximation is to match the obtained persistent set with every frame in the GOP. The approximated persistent set is generated for producing a good representation of the whole GOP, that means this set can be used for describing all frames in the GOP. Therefore, if the representation is of good quality, the majority of descriptors in each frame would find matches in the approximate descriptor set. The process of this experiment in each GOP is shown follows. Also, the size of GOP is K.



(a) GOP No.1 (b) GOP No.2 (c) GOP No.3

(d) GOP No.4 (e) GOP No.5 (f) GOP No.6

Figure 3.7: The results of matching the approximate set with each frame in the GOP when $K = 5$.

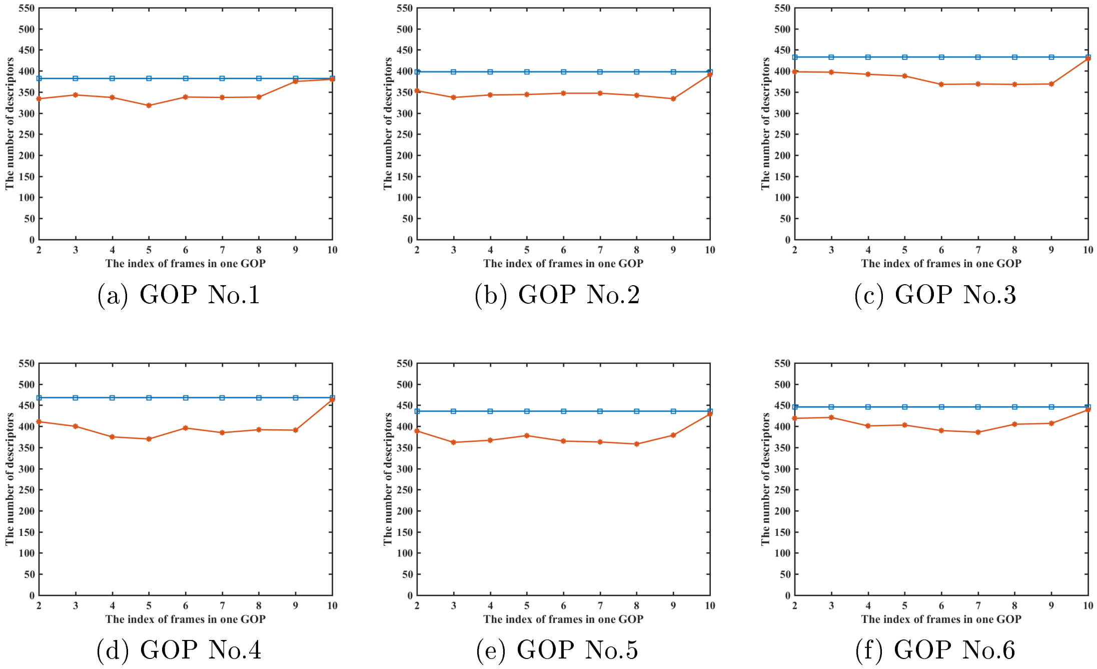
1. Approximate persistent descriptors by the approximation method. The obtained descriptor set is denoted by $S_{per-approx-1}$ with size M.

2. Extract SIFT descriptor sets from all frames in the GOP. Define the descriptor sets as S_{f_i}, i is the index of frame in the GOP, where $i \in \{1, 2, \cdots, K\}$.

3. Match $S_{per-approx-1}$ with all S_{f_i}(except for S_{f_1} since $S_{per-approx-1}$ is the subset of S_{f_1}) individually by ratio test and geometric consistency check, the numbers of descriptors from S_{f_i} that can find matches in $S_{per-approx-1}$ are denoted by N_i, where $i \in \{1, 2, \cdots, K\}$. This subset is defined as $S^{f_i}_{match}$, $i \in \{1, 2, \cdots, K\}$.

4. Compare N_i with M, where $i \in \{1, 2, \cdots, K\}$.

The comparison results are illustrated in Fig. 3.7, Fig. 3.8 and Fig. 3.9. These



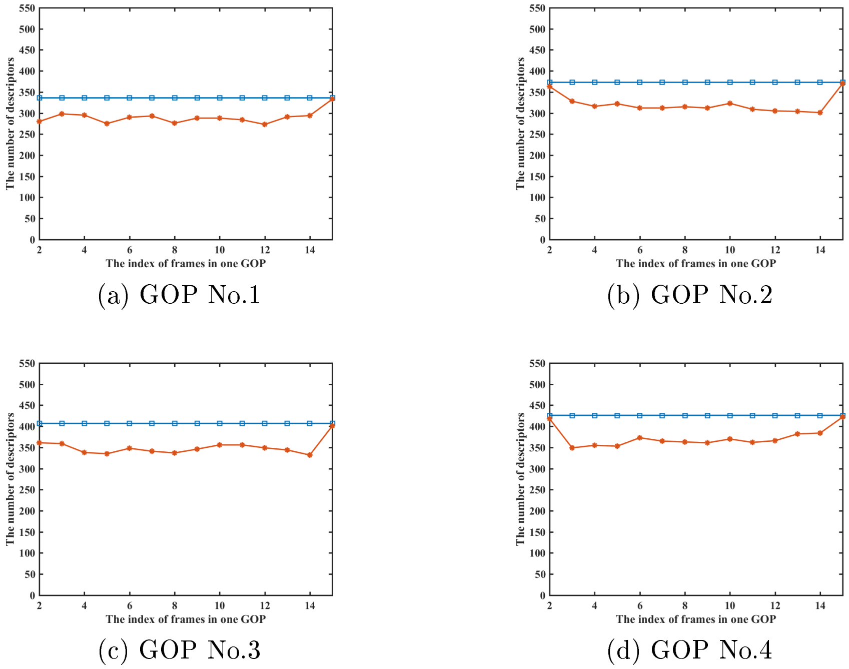
(a) GOP No.1 (b) GOP No.2 (c) GOP No.3

(d) GOP No.4 (e) GOP No.5 (f) GOP No.6

Figure 3.8: The results of matching the approximate set with each frame in the GOP when $K = 10$.

experiments are conducted on the same 60-frames video sequence as those tables shown in Table 3.1 to Table 3.3. In each sub-figures in Fig. 3.7 to Fig. 3.9, the blue line refers to the size of the approximate descriptor set $S_{per-approx-1}$ in the GOP, and the red line indicates the set of descriptors from each frame that can find matches in $S_{per-approx-1}$, that are, $S_{match}^{f_i}$, where $i \in \{1, 2, \cdots, K\}$. Note that the x-axis is the index of frames in the GOP, starting from 2. In each GOP, two things need to be pointed out:

1. The size of $S_{match}^{f_i}$ of each frame is close to the size of $S_{per-approx-1}$, where $i \in \{1, 2, \cdots, K\}$. As we can see from all figures in Fig. 3.7, Fig. 3.8 and Fig. 3.9, the red lines are always close to the blue lines, demonstrating that most descriptors in $S_{per-approx-1}$ can find matches in each frame within the



(a) GOP No.1

(b) GOP No.2

(c) GOP No.3

(d) GOP No.4

Figure 3.9: The results of matching the approximate set with each frame in the GOP when $K = 15$.

GOP. Thereby, $S_{per-approx-1}$ can offer a good representation of each frame in the GOP.

2. Compared to the last frame, intermediate frames have less descriptors that can find matches in $S_{per-approx-1}$. In most of the cases in the listed figures, the middle part of the red lines are a little bit far way from blue lines, meaning that $S_{per-approx-1}$ performs slightly poorer in representing the intermediate frames. As we know from keyframe labelling stage, keyframes are the first and the last frame in the GOP. Also, $S_{per-approx-1}$ is obtained from matching these two frames. Thus, more descriptors in the keyframes can find matches in $S_{per-approx-1}$. For the intermediate frames, some descriptors in $S_{per-approx-1}$ may not exist in their descriptor sets, resulting in a smaller matching set.

Two conclusions are drawn in these two experiments. One is that the proposed persistent descriptors approximation approach performs well to cover almost all persistent descriptors obtained by the direct method. Furthermore, it can provide a good representation for each frames in the GOP. Therefore, the proposed approach can be viewed as a nice way for persistent descriptors approximation.

However, although this approximation approach utilizes the temporal information of video sequence, it fails to exploit motion correlation, which is one of the inherent properties of video sequences. In the next section, instead of using geometric consistency check for removing outliers, block-motion estimation is performed to filter out wrong matches. It is another method to approximate persistent descriptors.

3.3.3 The Second Approximation Approach

In Section 3.3.2, it is verified that only processing two keyframes in a GOP can roughly provide a good representation of the contents in that GOP. However, how to use the intermediate frames to enhance the performance and how to use motion information should be considered. In this section, another persistent descriptor approximation approach that makes use of these two aspects together is presented.

Geometric consistency check is used to discard wrong matches after ratio test. Most of the time, those outliers also do not comply with the motion field in the video. If the intermediate frames in the GOP are exploited to estimate motion field, those outliers that do not obey the estimated motion field can be found and then be filtered out. Therefore, in some sense, motion field estimation can be an alternative to geometric consistency check stage. Block-motion estimation is used here to estimate the motion of descriptors.

Using video motion information to distinguish bad matches is the core of this approximation approach. Similar to the approach discussed in Section 3.3.2, the proposed approach in this section also starts from matching two keyframes in the GOP: SIFT descriptors are extracted from these two keyframes and ratio test is then used to generate matches between them. Instead of applying geometric consistency check, by using block-motion estimation, the location of each matching descriptor in the first keyframe is tracked across all the frames until it reaches to the last frame in the GOP. Therefore, each matching descriptor in the first keyframe has an associated estimated location in the second keyframe. By calculating the Euclidean distance between the location of the estimated location and its actual match in the second keyframe, it can be cleared that whether this descriptor obeys the estimated motion field or not. Specifically, if the obtained distance is small, it can be claimed that this descriptor complies with the motion field of the video sequence. On the other hand, if the distance is large, it means that this descriptor does not obey the estimated motion field. This descriptor thereby with high probability is not a correct match and will be removed. The detailed process to generate persistent descriptors in the GOP using this approximation approach goes through the following steps (note that the GOP size is K):

1. Extract two sets of descriptors, S_{kf1} and S_{kf2} from the first keyframe f_{kf1} and the second keyframes f_{kf2} in the GOP, respectively. These two sets can be of different sizes.

2. Match S_{kf1} and S_{kf2} using ratio test. Unique pairs of matches between S_{kf1}

and S_{kf2} are given by

$$S_{match} = \{(d_1^{kf1}, d_1^{kf2})_1, (d_2^{kf1}, d_2^{kf2})_2, \cdots, (d_{N_2}^{kf1}, d_{N_2}^{kf2})_{N_2}\},$$

where $d_i^{kf1} \in S_{kf1}$,$d_j^{kf2} \in S_{kf2}$ and $i, j \in \{1, 2, \cdots, N_2\}$. Note that the number of matching pairs, N_2, must be smaller or equal to the size of S_{kf1} or the size of S_{kf2}. All d_i^{kf1} form a descriptor subset D_{match}^{kf1} while all d_j^{kf2} form a descriptor subset D_{match}^{kf2}, where $i, j \in \{1, 2, \cdots, N_2\}$.

3. Estimate the location of each descriptor in D_{match}^{kf1} in the second keyframe.

   Define the location of d_i^{kf1} as l_i^{kf1} and the locations of d_j^{kf2} as l_j^{kf2}, where $i, j \in \{1, 2, \cdots, N_2\}$. Therefore, all l_i^{kf1} form a set L^{kf1} that consists of all the locations of descriptors in set D_{match}^{kf1}. Meanwhile, all l_j^{kf2} form a set L^{kf2} that consists of all the locations of descriptor set D_{match}^{kf2}. Then, the location of each descriptor from D_{match}^{kf1} in the second keyframe is estimated across all the intermediate frames using block-motion estimation. The detailed procedures of the method are given in Chapter 2. Thus, each descriptor d_i^{kf1} will have an estimated location l_{i-est}^{kf2} in the second keyframe, where $i \in \{1, 2, \cdots, N_2\}$. All these predicted locations form a set L_{est}^{kf2}.

4. For each descriptor d_i^{kf1}, calculate the Euclidean distance between the estimated location and the location of its match in the second keyframe. To be more specific, for each descriptor d_i^{kf1} in the first keyframe, there is a corresponding match d_j^{kf2} in the second key frame. The locations of these two descriptors are l_i^{kf1} and l_j^{kf2}, respectively, and $i = j$. Also, from the third step, there is an estimated location of l_i^{kf1}, l_{i-est}^{kf2}. Calculate the Euclidean distance between l_j^{kf2} and

Table 3.4: The comparison of $S_{per-approx-2}$, S_{dir}, $S_{match-2}$ and match percentage when $K = 5$

GOP index	size of $S_{per-approx-2}$	size of S_{dir}	size of $S_{match-2}$	match percentage
GOP No.1	208	210	102	48.57%
GOP No.2	208	199	97	48.74%
GOP No.3	218	147	77	52.38%
GOP No.4	217	165	75	45.45%
GOP No.5	233	259	111	42.85%
GOP No.6	235	225	111	49.33%
average	220	201	96	47.89%

l_{i-est}^{kf2}, denoted as $dist_i$, where $i \in \{1, 2, \cdots, N_2\}$. By applying this process to all descriptors in D_{match}^{kf1}, all Euclidean distances between the estimated locations and the locations of the matches in the second keyframe can be obtained.

5. Remove those matches with a large distance. This is done using a threshold $dist_{thre}$. For all $dist_i$ where $i \in \{1, 2, \cdots, N_2\}$, if $dist_i > dist_{thre}$, the matching pair $(d_i^{kf1}, d_i^{kf2})_i$ is removed since this pair does not obey the estimated motion field. The rest of the matches form the set $S_{per-approx-2}$. Note that descriptors in $S_{per-approx-2}$ are taken from the first keyframe.

There are two advantages for this new approximation approach. The first is that it is simpler than the method presented in Section 3.3.2. As discussed in Chapter 2, geometric consistency check is complicated in computation. The proposed approximation approach replaces geometric consistency check with motion field estimation to address the complexity issue. Secondly, the intermediate frames in the GOP are utilized in the motion estimation. Thus, the approximated persistent descriptors cover

Table 3.5: The comparison of $S_{per-approx-2}$, S_{dir}, $S_{match-2}$ and match percentage when $K = 10$

GOP index	size of $S_{per-approx-2}$	size of S_{dir}	size of $S_{match-2}$	match percentage
GOP No.1	126	92	35	38.04%
GOP No.2	113	67	25	37.3%
GOP No.3	118	108	38	35.18%
GOP No.4	126	87	29	33.33%
GOP No.5	121	100	26	26.00%
GOP No.6	119	65	21	32.3%
average	121	85	29	33.69%

Table 3.6: The comparison of $S_{per-approx-2}$, S_{dir}, $S_{match-2}$ and match percentage when $K = 15$

GOP index	size of $S_{per-approx-2}$	size of S_{dir}	size of $S_{match-2}$	match percentage
GOP No.1	87	38	17	44.73%
GOP No.2	82	19	3	15.79%
GOP No.3	85	40	11	27.50%
GOP No.4	86	41	9	21.95%
average	85	35	10	27.49%

more feature information from the GOP, indicating that these descriptors might produce a better representation for the GOP.

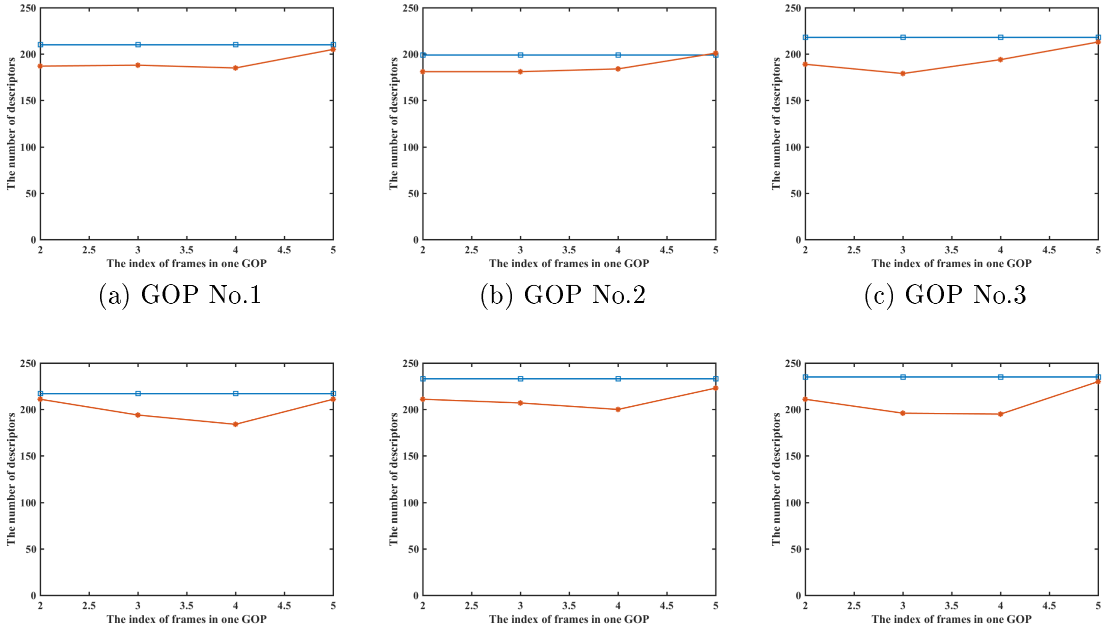
The two methods proposed in Section 3.3.2 can also be used here to measure the accuracy of this new approximation approach. Same experiments as these in Section 3.3 are carried out using this new approximation method. Also, in our examples shown in the rest of this section, the processed video sequence is the same 60-frames video.

The first approach to measure the performance of the approximation is to match the obtained descriptors with the descriptors obtained by the direct method. The results are shown in Table 3.4, Table 3.5 and Table 3.6. The GOP size are $K = 5$, $K = 10$ and $K = 15$, respectively. And the threshold for distance $dist_{thre}$ is fixed as 10.

As shown in these three tables, there are two observations that are worth analyzing:

1. The matching percentages are all below 50%. As we can see from these three tables, the match percentages are relatively low. For example, in Table 3.6, in GOP No.2, only 15.79% descriptors in S_{dir} can find matches in the approximation set $S_{per-approx-2}$. It indicates that $S_{per-approx-2}$ obtained by the approximation method proposed in this section is unable to contain that many descriptors obtained by the direct method. Therefore, the proposed approximation method does not perform well in capturing persistent descriptors.

2. As the GOP size increases, the size of approximation persistent set decreases significantly. As it is illustrated in Table 3.4, when $K = 5$, the average size of $S_{per-approx-2}$ is 220. The average size of $S_{per-approx-2}$ drops to 121 when K



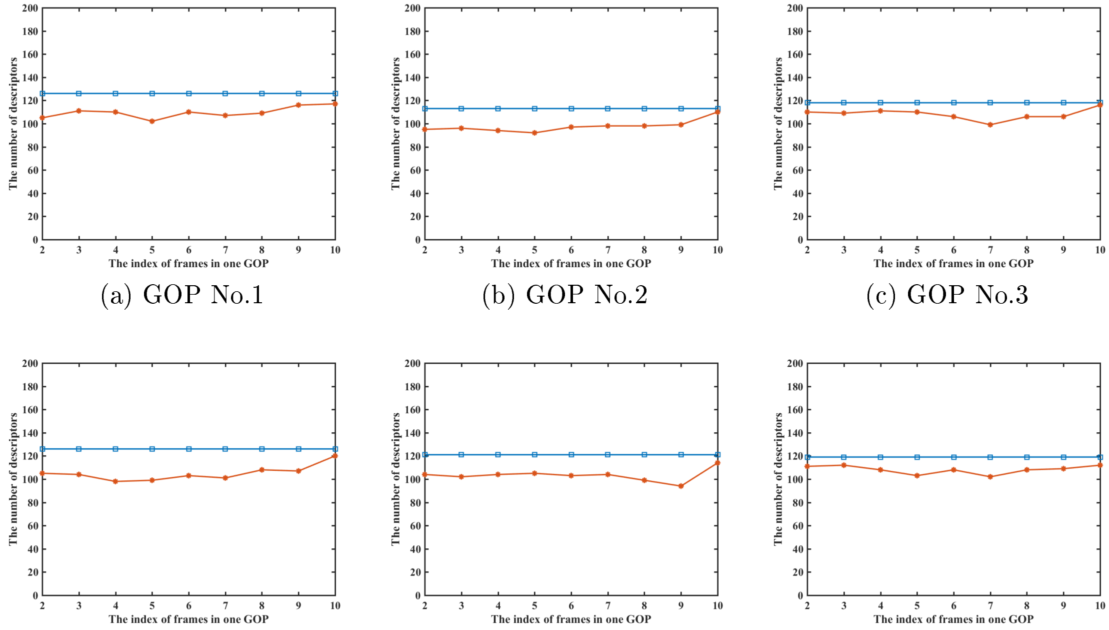
(a) GOP No.1 (b) GOP No.2 (c) GOP No.3

(d) GOP No.4 (e) GOP No.5 (f) GOP No.6

Figure 3.10: The results of matching the approximate set with each frame in the GOP when $K = 5$.

increases to 10, and it falls to 85 when $K = 15$. Therefore, descriptors tends to have large displacements that exceed the preseted threshold $dist_{thre}$ when the size of GOP increases.

The second measure approach is to distinguish whether the approximated set can produce a nice representation for the GOP. Following the same procedures shown in Section 3.3.2, by matching $S_{per-approx-2}$ with all intermediate frames in the GOP except for the first frame, the number of descriptors in each frame that can find matches in $S_{per-approx-2}$ can be obtained. This can be used to judge the performance of representation. The results are depicted in Fig. 3.10, Fig. 3.11 and Fig. 3.12. From these three figures, here comes two similar observations to those in Section 3.3.2. First, most descriptors in the approximated set can find matches in each



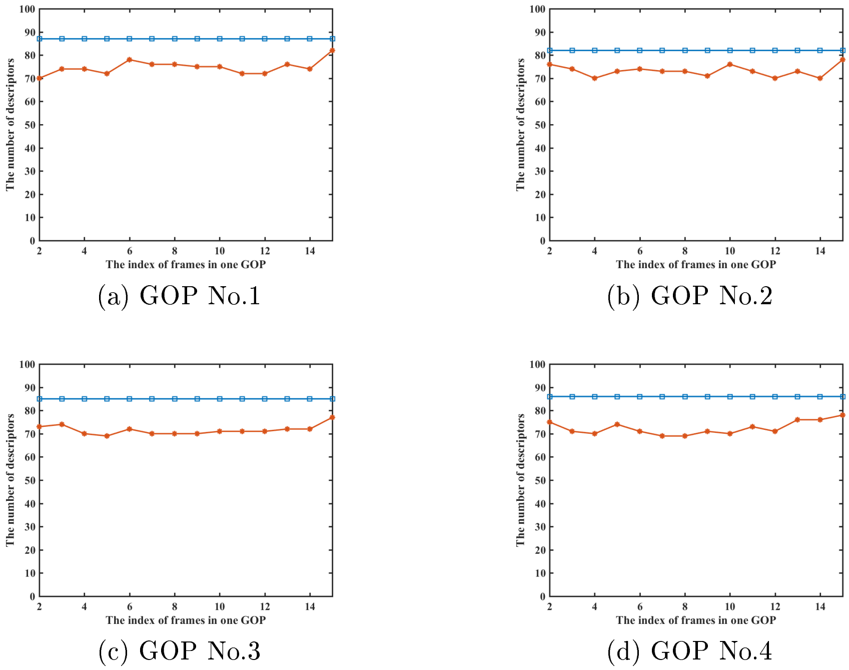
(a) GOP No.1 (b) GOP No.2 (c) GOP No.3

(d) GOP No.4 (e) GOP No.5 (f) GOP No.6

Figure 3.11: The results of matching the approximate set with each frame in the GOP when $K = 10$.

frame in the GOP. In this sense, the approximated descriptor set can generate a good description for the feature information of the GOP. Secondly, compared to the keyframes, the intermediate frames in the GOP find less matches in $S_{per-approx-2}$. It is reasonable because descriptors in $S_{per-approx-2}$ are derived from the subset of matching the two adjacent keyframes, which are the first frame and the last frame in the GOP. Therefore, the last frame in the GOP can find more matches in $S_{per-approx-2}$ than the intermediate frames.

By analyzing the results of these two different performance measures, we conclude that the proposed approximation approach behaves well for providing a compact representations of each frames in the GOP. Furthermore, it utilizes the intermediate frames in the GOP to predict the estimated motion field for bad matches. However,



(a) GOP No.1

(b) GOP No.2

(c) GOP No.3

(d) GOP No.4

Figure 3.12: The results of matching the approximate set with each frame in the GOP when $K = 15$.

it is weak to cover majority of persistent descriptors in the GOP, which might be a problem in the pairwise-matching stage.

3.4 Persistent Descriptors Selection

Three different persistent descriptors extraction approaches are presented in Section 3.3.1 to Section 3.3.3. It is obvious that the size of the persistent set would change when the video content and motion speed of the video sequence vary. Even in the same video, different GOPs might produce different numbers of approximated persistent descriptors. Therefore, in order to generate a size-controlled persistent set for each GOP, the selection stage is crucial. Another important reason for descriptor selection

Table 3.7: Matching results in random selection

reference frame index	test 1	test 2	test 3
frame 1	26	34	35
frame 2	27	35	36
frame 3	27	36	38
frame 4	27	38	34
frame 5	30	39	38
average	27.4	36.4	36.2

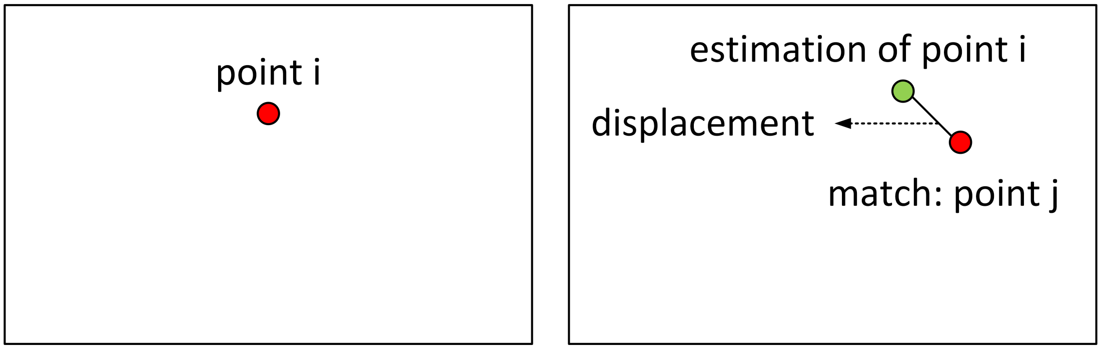
is that different approximated persistent descriptors perform differently when they are used for matching with reference videos. This statement is tested by a simple experiment. Assume there is a GOP from the query video and persistent descriptors are extracted using any approach described in Section 3.3.1 to Section 3.3.3. There is another reference video, and the goal is to do matching between the persistent set and the descriptors extracted from each frame in the reference video. In each experiment run, a fixed number of descriptors are randomly selected from the persistent set. If the matching result varies every time, conclusion can be drawn that even in the same persistent set, different descriptors contribute differently to pairwise-matching.

The result is shown in Table 3.7. In this experiment, the first approximation method presented in Section 3.3.2 is used to create a set of persistent descriptors from the GOP. 80 descriptors are randomly selected from this set out of the total number of descriptors in the set. We know that the ground truth of the query GOP and the reference video is a match pair. By matching the selected persistent descriptors with all the extracted SIFT descriptors in these five frames using both ratio test and geometric consistency check, the number of persistent descriptors that find matches in each reference frames is obtained.

As can be seen from the table, in the first test, for the first reference frame, 26 out of 80 persistent descriptors can find matches. However, in the second test, the number of matching descriptors is 34. It turns out that the numbers of matches for different reference frames fluctuate. This can be explained by the randomness in selecting in the persistent set. We know that random selection means that the selected descriptors are not exactly the same each time. From these results, even for the same approximated persistent descriptor set, different selected descriptors have different matching performance. The similar situation happen in the other tests. The average number of matches for each test also shows the difference in these three tests. Therefore, it is clear that favoring different descriptors within the persistent set greatly influences the behavior of pairwise-matching. In order to select those descriptors that are more likely to be matched in pairwise-matching, persistent descriptors selection component must be included in the encoding scheme.

In this section, three different selection approaches are presented to address the descriptor selection problem, they are: selection based on displacements, selection based on clusters and selection based on blocks. The detailed description of these selection methods is given in the following subsections. Like persistent descriptors extraction, the selection stage also operates on GOP level. The number of selected descriptors depends on the data rate budgets. Usually, there are several different operating modes.

In order to discuss these selection methods, the persistent descriptors extraction method is fixed. Here, the method discussed in Section 3.3.2, which is matching two keyframes in the GOP by ratio test and geometric consistency, is applied for persistent descriptors approximation. The obtained persistent set is $S_{per-approx-1}$. What needs



(a) The first keyframe (b) The second keyframe

Figure 3.13: The illustration of displacement definition.

to be emphasized is that those selection methods can be applied to any extraction approach. In the following subsections, the number of selected descriptors is denoted as N_{select}.

3.4.1 Selection Based on Displacements

As mentioned in Section 3.3.2, in the extraction stage, estimated motion field can be an alternative to geometric consistency check. However, motion prediction cannot only be used in persistent set extraction, but it also can be exploited in the selection stage. The rule for the selection is based on displacements.

The definition of displacement is illustrated in Figure.3.13. Assume that point i is the location of a persistent descriptor in $S_{per-approx-1}$. The location of the corresponding match in the second keyframe is point j, which is the green point in Fig. 3.13b. By predicting the location of point i in the second keyframe using block-motion estimation, an estimated location for point i is obtained. Then, the displacement is defined as the Euclidean distance between the estimated location of point i and the location of point j in the second frame. Displacement is a parameter for measuring the extent of the descriptor obeying the motion field. Suppose the estimation of motion field is perfect. Smaller displacement means that the descriptor is more likely to be better in terms of motion information consistency.

Following this line, the detailed procedure of selection within a GOP based on displacements is described as follows:

1. Estimate the locations of all the approximated persistent descriptors in $S_{per-approx-1}$ in the second keyframe.

2. Compute the displacements of all the descriptors in the first stage.

3. Rank all the displacements in descending order. The descriptors with smallest N_{select} displacements are selected.

Therefore, by measuring displacements, a set of descriptors in $S_{per-approx-1}$ is selected. In terms of displacement ranking, these descriptors with higher ranks are more likely to be complaint with the estimated motion field. In this sense, the rank reflects the confidence in what we select.

In summary, the proposed selection method based on displacements makes use of motion information by using motion field prediction. By calculating the difference between the estimated location and the location of its actual match, the displacement of each descriptor is obtained for ranking descriptors.

3.4.2 Selection Based on Clusters

As discussed in Section 1.2, some novel schemes of video descriptor coding are designed by applying clusterings to video descriptors. Those schemes do clustering to descriptors by using NMF, K-means and so on. Inspired by their work, the idea of

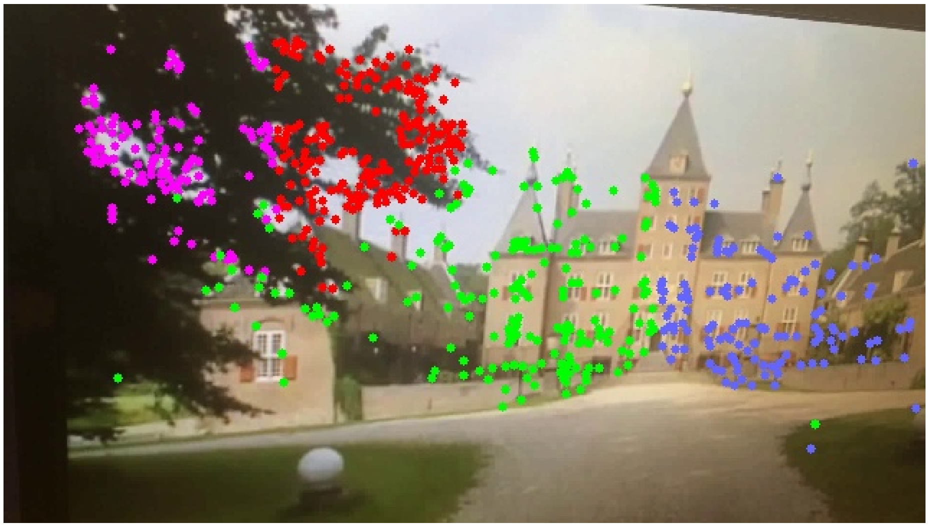
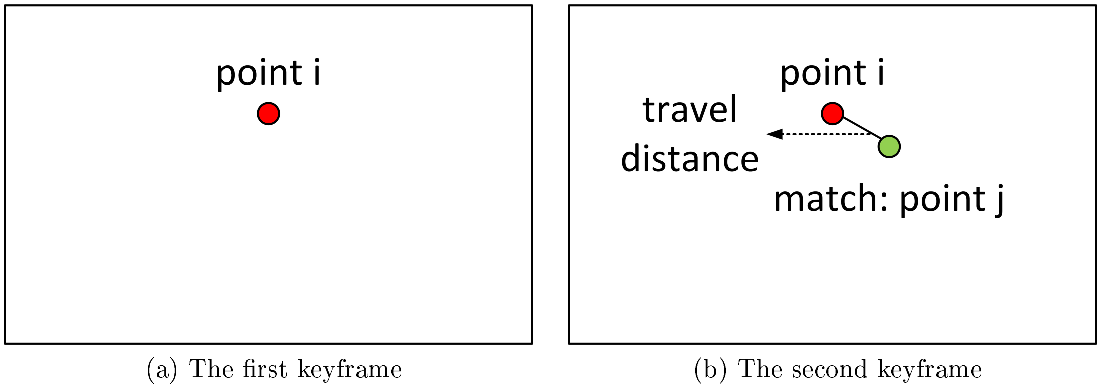


Figure 3.14: An example of frame clusters.

selection based on clustering arises. Instead of doing clustering based on the feature part of descriptors, our approach mainly focuses on doing clustering of the descriptors locations. In order to make the process easier, K-means algorithm is applied for location classification due to its efficiency [39]. In the experiments shown in Chapter 5, K-means is implemented by using the function that is available in *OpenCV* library.

An example of the frame clusters is displayed in Fig. 3.14. Here, the number of clusters $K_{cluster}$ is fixed as $K_{cluster} = 4$. As shown in Fig. 3.14, each point is the location of a descriptor in $S_{per-approx-1}$, and four different colors represent different clusters. Thereby, after clustering, the descriptors selection problem is transfered from the whole frame to each cluster.

Another important problem is to select descriptors within each cluster. It is very straightforward to utilize the motion information, which is a significant property of videos. Nevertheless, instead of making use of the estimated motion field, the actual movement is exploited, which is named by travel distance. A simplified illustration



(a) The first keyframe (b) The second keyframe

Figure 3.15: Illustration of the travel distance definition.

of the travel distance is shown in Fig. 3.15. Point i represents the location of a descriptor from $S_{per-approx-1}$. The location of its corresponding matching descriptor in the second keyframe is point j. By calculating the Euclidean distance between the location of point i and the location of point j, the distance that the persistent descriptor travels across all the frames in the GOP is obtained. The travel distance becomes a parameter to measure the quality of the descriptor.

Within each cluster, the descriptors with smaller travel distances are favored. The reason is that compared with the descriptors with larger travel distances, descriptors with small travel distances are more likely to be stationary. Therefore, they are more likely to be persistent. The detailed procedure for selection based on clusters is described here:

1. Partition all the locations of descriptors in $S_{per-approx-1}$ into $K_{cluster}$ clusters by K-means algorithm.

2. Calculate the travel distance for each matching descriptor pair.

3. Rank descriptors within each cluster by the travel distance. The larger travel

distance is, the lower the descriptor is ranked.

4. Select the descriptors with higher rank in each cluster. The number of selected persistent descriptors in each cluster depends on the ratio of the number of the descriptors to its cluster over the size of $S_{per-approx-1}$. To be more specific, denote the number of descriptors in the current cluster as N_i, where $i \in \{1, 2, \cdots, K_{cluster}\}$ and i is the index of cluster and the size of $S_{per-approx-1}$ is denoted as N_{total}. It has $\sum\limits_{i=1}^{K_{cluster}} N_i = N_{total}$. The number of descriptors that are supposed to be selected from cluster i is given by

$$N_{select-i} = \left\lfloor \frac{N_i}{N_{total}} \times N_{select} \right\rfloor . \tag{3.2}$$

Therefore, by measuring the travel distances within each cluster, a set of descriptors from $S_{per-approx}$ is selected. In conclusion, the novelty of this proposed selection approach comes from two different aspects. On one hand, this method introduces the idea of clustering for descriptor location classification. Clusters can roughly describe how many interesting regions or objects in the frame. By selecting descriptors within each cluster, we can make sure that the selected descriptors can approximately cover all the features in the persistent set. On the other hand, the definition of the travel distance is applied for descriptor ranking in each cluster. Stationary descriptors are favored to guarantee that descriptors could obey the definition of descriptor persistency.

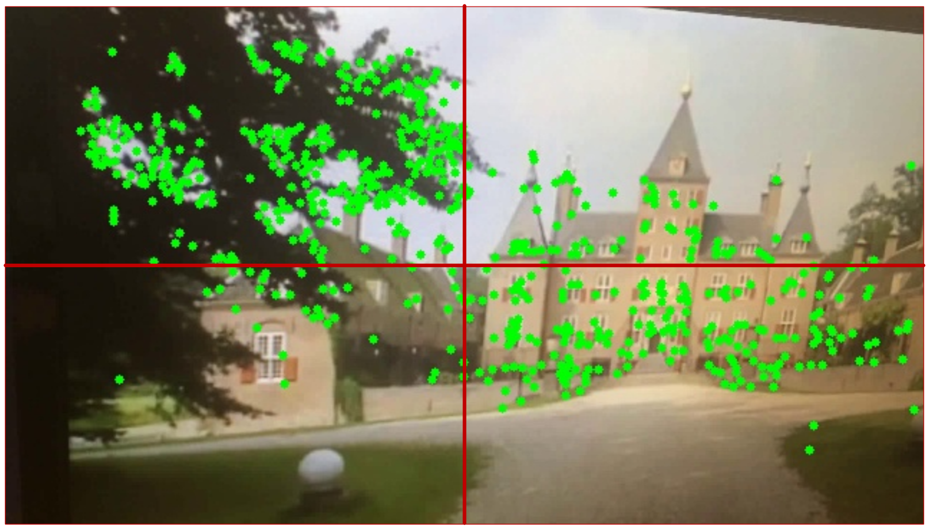


Figure 3.16: The partitioning method of the selection based on blocks.

3.4.3 Selection Based on Blocks

Another selection approach is designed based on blocks. The original motivation for generating such selection method is to cover as much information as possible for the whole frame. Compared with the clustering method, a simpler way is to partition all the descriptors into four blocks based on their locations. The partitioning method for the selection is displayed in Fig. 3.16. The green points are the location of descriptors in $S_{per-approx-1}$. The red lines illustrate the method for partitioning. In each GOP, the first keyframe is divided into four big blocks. These four blocks are equally sized. In this stage, the descriptor selection problem in the whole frame is converted to a selection problem within each block.

Different from the discussion in Section 3.4.2, the selection method has changed a little bit. Although within each block, the descriptors with smaller travel distances are still favored, the number of descriptors that should be kept in each block is different from that used in the method based on clusters. In this selection approach, in the block with sparse descriptors, more descriptors are selected since those descriptors are more likely to be important to capture the information in such block. For the blocks with dense descriptors, smaller number of descriptors in such set can still describe the feature information. Thus, less descriptors are required. The details about the selection method based on blocks is shown in Algorithm 1.

Algorithm 1: Selection algorithm based on blocks

- Partition all descriptors in $S_{per-approx-1}$ into four sets using the method shown in Fig. 3.16. These four sets are denoted by S_{SE}, S_{SW}, S_{NE} and S_{NW}.

- Find the sort index for the four sets in ascending order of their size. Denote the indices as $sort_1$, $sort_2$,$sort_3$, $sort_4$. The obtained descriptor sets are thereby S_{sort_i} with the denoted by N_{sort_i}, where $i \in \{1, 2, 3, 4\}$. The size of descriptors set respectively selected from these four sets are denoted as $N_{select-sort_i}$, where $i \in \{1, 2, 3, 4\}$.

- Compute all the travel distances of descriptors in each block.

- For S_{sort_1} and S_{sort_2} that are with the least number of descriptors and the second least, respectively, $N_{select-sort_i}$ number of descriptors with smaller travel distance are selected. Denote the ratio parameter as δ. $N_{select-sort_i}$ is computed by:

  $$N_{select-sort_i} = \min \left\{ \left\lfloor \left(\frac{N_{sort_i}}{\sum\limits_{j=1}^{4} N_{sort_j}} + \delta \right) \times N_{select} \right\rfloor, N_{sort_i} \right\},$$

  where $i \in \{1, 2\}$ and $\delta \in (0, 0.5)$.

- For S_{sort_3} and S_{sort_4}, the number of descriptors selected from each set is computed as:

  $$N_{select-sort_i} = \left\lfloor \left(N_{select} - \sum_{i=1,2} N_{select-sort_i} \right) \times \frac{N_{sort_i}}{\sum\limits_{j=3,4} N_{sort_i}} \right\rfloor.$$

  Also, the descriptors with smaller travel distances are kept.

Chapter 4

Decoding and Pairwise-Matching

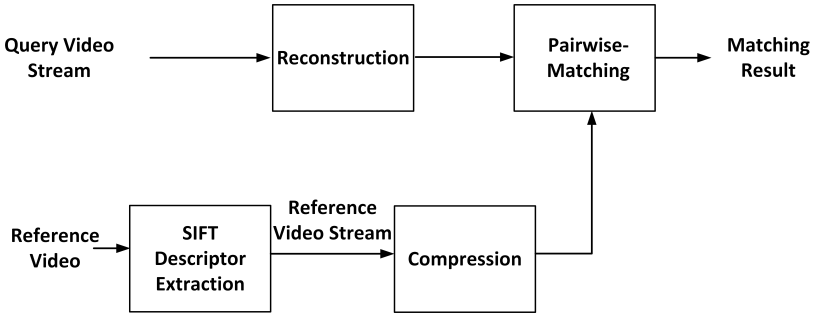


Figure 4.1: The block diagram of the proposed decoder and pairwise-matching.

In this chapter, a decoder designed for performing pairwise-matching of videos is proposed. The main components of the decoder are described in Section 4.1. In Section 4.2, a simple method for pairwise-matching is presented along with its pseudo-code.

4.1 Decoding

The block diagram of the proposed decoder and pairwise-matching is showed in Fig. 4.1. There are two parallel input streams that are fed to the decoder. Pairwise-matching is based on two videos, which are the query video and the reference video. Therefore, the first stream is the query video that is encoded using schemes described in Chapter 3. The other input stream is the reference video stream. This stream includes uncoded SIFT descriptors from all reference frames.

In the first stage of the decoder, the query video stream is processed to reconstruct several persistent sets. According to Chapter 3, the number of the persistent sets is determined by the number of GOPs in the query. Meanwhile, the reference video stream is quantized using the same quantization method as which is used for persistent descriptors. Note that there is no selection for the operations of reference video. All SIFT descriptors extracted from all frames in the reference video are preserved.

Now, these two processed streams are ready be fed to the pairwise-matching process. This is the second stage of the decoder. The detailed method for pairwise-matching is discussed in the next section.

4.2 Pairwise-Matching Method

We define two videos to be a matching pair if these two contain the same objects or describe the same content. A simple pairwise-matching method is presented in this thesis. That is, the query video and reference video are declared a match if more than half of the GOPs from the query video can find matches in the reference frames. The detailed pairwise-matching method is shown in Algorithm 2.

Algorithm 2: Pairwise-matching algorithm

```
Input:
    Query video stream: reconstructed persistent descriptor sets $S_{per-approx-i}$,
      where $i \in \{1, 2, \cdots, N_{GOP}\}$;
    Reference video stream: compressed reference descriptor set for each
      reference frame $S_{ref-j}$, where $j \in \{1, 2, \cdots, N_{ref}\}$;
Pairwise matching:
    $N_{match} = 0$;
    for $i = 1$ to $N_{GOP}$ do /* traversing all persisent sets */
        for $j = 1$ to $N_{ref}$ do /* traversing all reference frame
          descriptor sets */
            Match $S_{per-approx-i}$ to $S_{ref-j}$ using ratio test and geometric
              consistency check;
            Calculate matching score $score_{i,j}$ for these two sets;
        end
        Find the largest matching score $score_{i-max}$ over all $score_{i,j}$ as the
          matching score for $S_{per-approx-i}$;
        if $score_{i-max} > score_{thre}$ then
            $N_{match} = N_{match} + 1$;
        else
            The persistent set $S_{per-approx-i}$ cannot find a match in the reference
              video;
        end
    end
Decision Making:
    if $N_{match} > \frac{N_{GOP}}{2}$ then
        These two video are declared a match;
    else
        These two videos are declared not a match;
    end
```

To be more specific, assume there are N_{GOP} GOPs in the query video. Then, N_{GOP} sets of reconstructed persistent descriptors, $S_{per-approx-i}$, where $i \in \{1, 2, \cdots, N_{GOP}\}$ are generated in the reconstruction stage at the decoder side. Denote the number of frames in the reference video as N_{ref}. N_{ref} sets of compressed descriptors denoted by S_{ref-j} are obtained after compression of the reference video stream, where $j \in \{1, 2, \cdots, N_{ref}\}$. Denote the number of persistent sets that find matches as N_{match}. In order to match these two videos, descriptors from $S_{per-approx-i}$ where $i \in \{1, 2, \cdots, N_{GOP}\}$ are matched to those of every reference frame. For each operation between $S_{per-approx-i}$ and S_{ref-j}, where $i \in \{1, 2, \cdots, N_{GOP}\}$ and $j \in \{1, 2, \cdots, N_{ref}\}$, ratio test and geometric consistency are applied to generate matching scores in which the highest matching score is considered as the matching score for GOP i. If this highest matching score exceeds the prescribed threshold $score_{thre}$, it is declared that the persistent set $S_{per-approx-i}$ can find match in the reference video. Otherwise we declare this set cannot find a match in the reference video. After all GOPs are processed, if $N_{match} > \frac{N_{GOP}}{2}$, these two videos are declared a match.

Chapter 5

Experimental Results and Evaluation

This chapter assesses the overall performance of the proposed descriptors selection schemes. In Section 5.1, the experiment description is given. The datasets used for performance evaluation is introduced in Section 5.2. In the last section, the results of the experiments are discussed and analyzed.

5.1 Experiment Description

In order to evaluate the performance of videos pairwise-matching, two performance metrics defined in Evaluation Framework for CDVA [2] are adopted in this thesis, namely false positive rate (FPR) and true positive rate (TPR). These two rates are defined as follows:

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP{+}TN}}, \tag{5.1}$$

$$\text{TPR} = \frac{\text{TP}}{\text{TP+FN}}, \tag{5.2}$$

where TP is the number of true positives, FP is the number of false positives, TN is the number of true negatives and FN is the number of false negatives. These two rates are evaluated using three datasets, that are query set, reference set and distractor set. The detailed description of the datasets is given in the next section.

5.2 Description of Datasets

In this thesis, the datasets used for performance evaluation are from MPEG-CDVA database [1]. It is composed of more than 10000 videos that cover various categories, for instance, animals, people, books, buildings, cars and so on. Due to the limitations of our computational resources, we randomly selected 180 videos from this huge database as the query videos from six categories: animals, buildings, people, plants, books and food. The reference set is selected from the categories as those used for the query set. The ground-truth is that query videos and reference videos are matching video pairs. The distractor set contains videos that are non-matching to the query videos, i.e., cross category pairs. All videos in the three sets have a resolution of 640 × 360 and a frame rate of 30 frames per second (fps).

5.3 Performance

This section presents the results of several experiments that were run for performance evaluation. The performance of the direct method is given along with its drawbacks in Section 5.3.1. In Section 5.3.2, the performance of the first approximation method,

Table 5.1: Encoder and decoder configurations

GOP size K	10
Threshold $dist_{thre}$	30
Threshold $score_{thre}$	15
Ratio parameter δ	0.06

which extracts persistent sets using keyframes, ratio test and geometric consistency check, is evaluated. The results for the second approximation method introduced in Section 3.3.3 is given in Section 5.3.3.

In each subsection, three different selection methods are applied after approximation to control the bitrate. In order to test the performance of the selection methods, four operating modes are set in all experiments. These are 80 descriptors, 120 descriptors, 150 descriptors and 200 descriptors. An operating mode is a parameter representing the number of descriptors that are selected in each GOP.

The detailed encoder and decoder configurations are given in Table 5.1. $dist_{thre}$ is the threshold for displacements for the second approximation method. $score_{thre}$ is the matching score threshold for pairwise-matching. Ratio parameter δ is used in the selection method based on blocks.

The target false alarm rate is set to be equal or less than 1% according to [2]. That means, for TPR evaluation, the target for FPR is less or equal to 1%.

5.3.1 Performance of the Direct Method

As discussed in Section 3.3.1, the direct method is the most accurate method for identifying persistent descriptors in the GOP. Therefore, the performance evaluation for the direct method is very crucial. However, when it comes to experiments, we

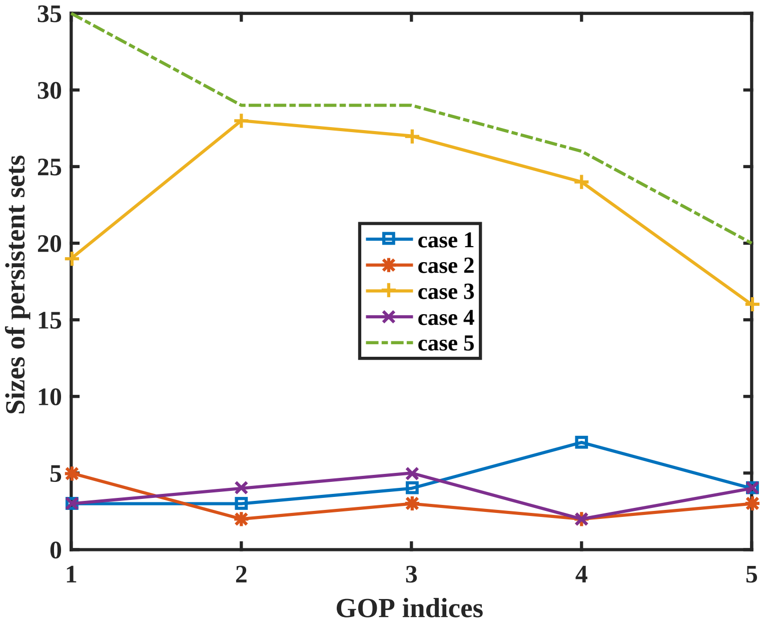


Figure 5.1: The sizes of persistent sets in five different-content video cases.

notice that when $K = 10$, there is only a few persistent descriptors that can be obtained in majority of GOPs. Moreover, the sizes of the persistent set in different GOPs significantly fluctuate. In order to verify this statement, five different-content videos are randomly picked from the query dataset. The direct method is applied to extract the persistent sets in these five videos. The sizes of persistent sets are shown in Fig. 5.1. Five GOPs in each video are processed in this example.

As shown in Fig. 5.1, two observations should be pointed out:

1. In all these five video sequences, within each GOP, the size of the persistent set is small. In Fig. 5.1, in case 1, 2 and 4, the number of the persistent descriptors in each GOP is less than 10. Among all the five videos, the largest persistent set is the one obtained from case 5, that is 35. Those numbers are much smaller than the first operating mode, 80 descriptors.

Table 5.2: TPRs for different Ks of the direct method

GOP size K	TPR (%) for $\leq 1\%$ FPR
5	81.01
8	70.39
10	65.36

2. For a video sequence, the numbers of persistent descriptors in different GOPs fluctuate. For example, in the first GOP of case 3, the number of persistent descriptors is 19. It increases to 28 in the second GOP, which is a big jump. In the following two GOPs, the sizes of persistent set slightly drop. It decreases to 16 in the fifth GOP. The similar situation also happens to the other four videos.

From observation 1, we expect that majority of GOPs in the query dataset fail to produce more than 80 descriptors. Therefore, selection stage is no longer necessary when the direct method is used to find persistent descriptors. The result is given in Table 5.2, where TPR = 65.36% when $K = 10$. Even if all the extracted persistent descriptors are used in pairwise-matching, the performance is still relatively poor. The reason for this could be attributed to the small sizes of persistent sets generated by the direct method. From Chapter 2, it is known that more matching descriptor pairs will produce higher matching score for two video frames. If a GOP can not produce a reasonable number of persistent descriptors, it is obvious that the obtained matching score is relatively low. Therefore, smaller sizes of persistent sets might result in lower TPR in pairwise-matching process.

In order to further analyze the behavior of the direct method, two more results for $K = 5$ and $K = 8$ are listed in Table 5.2. As we can see, TPR is 81.01% when $K = 5$, which is much better than the case when $K = 10$. As K increases to 8, TPR decreases to 70.39%. Although it is still better than the TPR for $K = 10$, there is more than 10% gap between $K = 5$ and $K = 8$.

From the previous analysis, it turns out that when K decreases, the performance of the direct method gets much better. However, the sizes of some persistent sets exceed the first operating mode, indicating that the bitrate can not be controlled. Due to the uncontrollability of the number of descriptors in each persistent set, we cannot totally rely on the direct method. In order to make use of the direct method, an adaptive way to control the sizes of the persistent sets is required. When the GOPs in a video cannot produce a reasonable number of persistent descriptors, the GOP size needs to be decreased in order to guarantee more persistent descriptors are generated. However, this is a challenging task since it will increase the computational complexity of the encoder. On the other hand, the decoder requires extra information on the different GOP sizes, which makes the communication more complex.

5.3.2 Performance When the First Approximation Method is Used

In Section 3.3.2, the performance of the first approximation method is evaluated using two experiments. The first experiment tests whether the first approximation method can cover most of the persistent descriptors obtained by using the direct method. The results from Section 3.3.2 show that the first approximation method performs well in capturing persistent descriptors. On the other hand, the second experiment shows that the first approximation method also gives a good representation for all frames in the GOP. Most of the approximated persistent descriptors can find matches in the intermediate frames. Therefore, the first approximation method performs well in persistent descriptors approximation.

Table 5.3: The results of different selection methods with the first approximation approach

Selection	TPRs of different Operating modes for $\leq 1\%$ FPR			
	80 Desc.	120 Desc.	150 Desc.	200 Desc.
Displacements	80.56%	86.67%	86.67%	87.22%
Clusters	80.56%	85.56%	86.11%	87.22%
Blocks	82.22%	86.11%	86.67%	87.22%

The performance of different selection methods are then evaluated based on the first approximation method. Table 5.3 depicts the results of different operating modes when different selection methods are applied.

In Table 5.3, there are several observations:

1. When the operating mode is 80 descriptors, the selection method based on blocks outperforms the other two selection methods. As it can be seen from Table 5.3, when 80 descriptors are selected from every GOP, the TPR of selection based on blocks reaches 82.22%. However, in the other two methods, TPRs are 80.56%, which is about 1.7% less than the best method.

2. When the operating mode is larger than 80 descriptors, the performance of these three selection methods tends to be the same. For example, in the case of 150, the block-based selection performs as good as the selection based on displacements while the result of the cluster-based method is 0.5% less than those of the other two. When the operating mode increases to 200, these three methods have the same performance, which is 87.22%.

3. In general, selection based on clusters performs slightly worse than the other two methods. Foe example, in the case of 120, selection based on clusters achieves

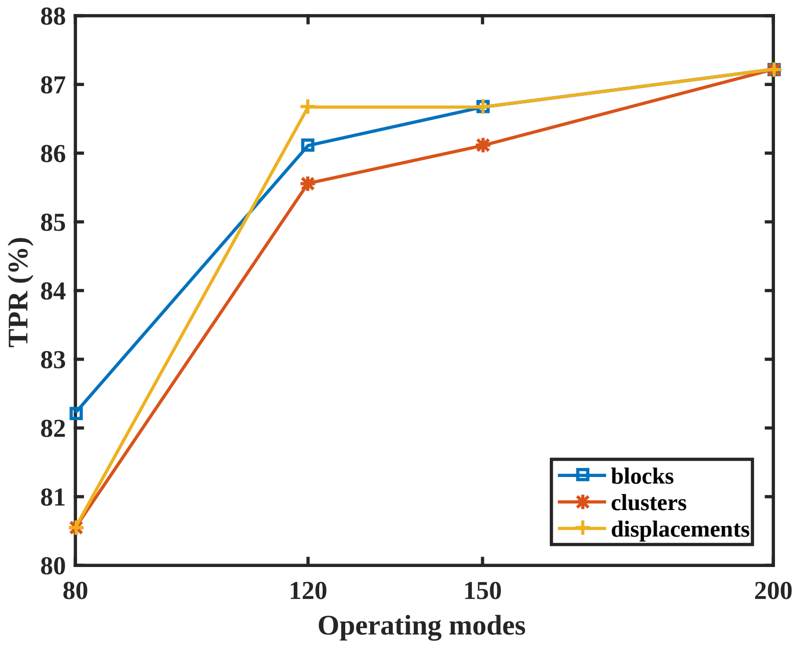


Figure 5.2: The results of selection methods under different operating modes with the first approximation method.

a TPR of 85.56%, which is 1% less than the other two selection methods.

4. The increase of TPR is nonlinear as the number of selected descriptors rises. Fig. 5.2 show the correlation between TPR and the number of selected descriptors. When the operating mode increases from 80 to 120, all TPRs of the three selection methods rise dramatically. However, as the operating mode increases to 150, the change of TPR is smaller compared to the previous change.

In conclusion, when the operating mode is 80, the method of selection based on blocks outperforms the other two methods. However, as the number of persistent descriptors increases, these three selection methods tend to have similar performance.

Table 5.4: The results of different selection methods with the second approximation approach

Selection	TPRs of different Operating modes for ≤ 1% FPR			
	80 Desc.	120 Desc.	150 Desc.	200 Desc.
Displacements	51.67%	56.67%	59.44%	60.55%
Clusters	51.67%	55.56%	58.33%	60.55%
Blocks	51.11%	55.00%	58.33%	60.55%

5.3.3 Performance When the Second Approximation Method is Used

In Section 3.3.3, the descriptors obtained by the second approximation method are compared to those obtained by the direct method. The results show that this approximation approach fails to capture the majority of persistent descriptors produced by the direct method. However, it is still capable of providing a good representation of each frame in the GOP.

For the second approximation method, several experiments were conducted using the three different selection methods and the results are shown in Table 5.4. From this table, several observations can be obtained:

1. In general, selection based on displacements outperforms the other two methods. When the operating mode is 80, selection based on displacements produces a TPR of 51.67%, which is the same as selection based on clusters. Then, in both 120 descriptors and 150 descriptors cases, the results of selection based on displacements are the highest among those of all three methods.

2. When operating mode is 200 descriptors, three selection methods produce the same TPR. In the last operating mode, all three selection methods produce the

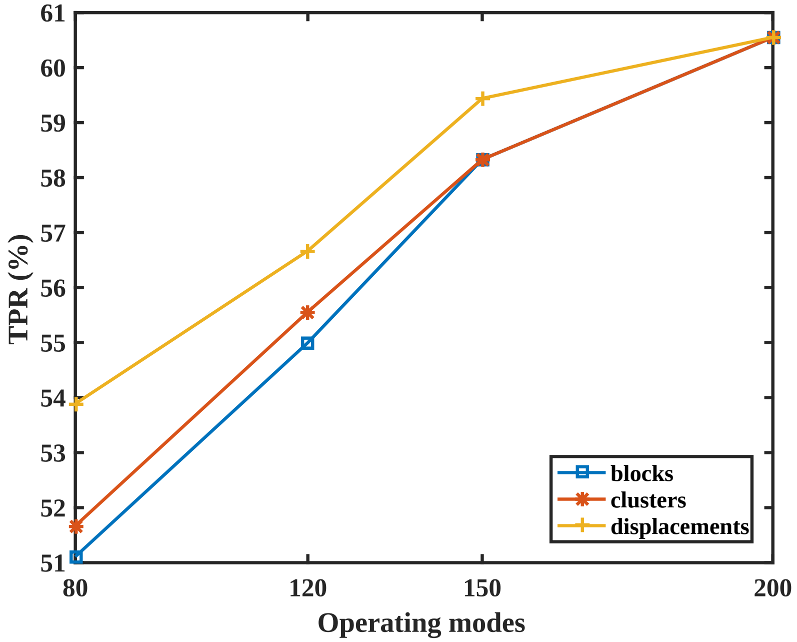
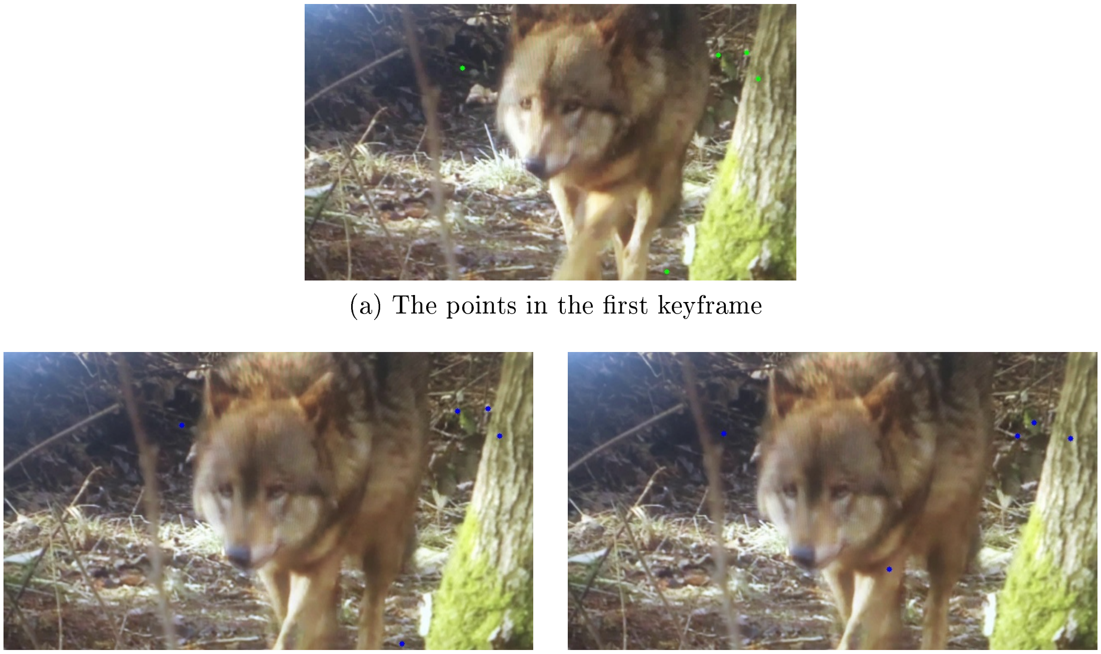


Figure 5.3: The results of selection methods under different operating modes with the second approximation method.

same TPR that is 60.55%.

3. The increase of TPR is nonlinear as the number of selected descriptors rises. Fig. 5.3 shows the trade-off between TPR and the size of persistent set. As shown in the figure, for any selection method, TPR increases approximate linearly when operating mode rises from 80 to 150. However, when the operating mode increases from 150 to 200, the increases are subtle, which indicates that there is less gain when the size of persistent set increases after the case 150.

4. This approximation method performs poorly compared to the first approximation method. If we compare Table 5.3 and Table 5.4, it turns out that the first approximation method produces much higher TPRs no matter which selection



(a) The points in the first keyframe

(b) The locations of actual matches in the second keyframe

(c) The estimated locations of the points in the second keyframe

Figure 5.4: An example of the inaccuracy in block-motion estimation.

method is used.

One of the main reasons why this approximation method performs poorly is that block-motion estimation might cause some errors in motion tracking. Although block-motion estimation is straightforward, it is sensitive to some variations, such as viewpoints, scaling and camera motion. Therefore, it is challenging for block-motion method to produce accurate estimations all the times. When displacements are calculated to discard some descriptors in the first keyframes, some of the correct matches would be removed since the obtained displacements exceed the prescribed threshold $dist_{thre}$.

Fig. 5.4 illustrates an example of the inaccuracy of block-motion approximation. Five descriptors in the first keyframe are showed in green in Fig. 5.4a. Fig. 5.4b gives the actual matches of these five descriptors in the second keyframe. The matches are obtained by ratio test and geometric consistency. Fig. 5.4c shows the estimated locations of these descriptors in Fig. 5.4a. By comparing locations of these five descriptors in Fig. 5.4b and Fig. 5.4c, it is clear that there is at least one incorrect estimation that locates at the bottom of the frame.

5.4 Comparison with Existing Schemes

A fair and thorough comparison with the existing schemes is somewhat difficult because none of the existing schemes only focus on video descriptors selection problem. However, to the best of our knowledge, the simulation results indicate that our scheme outperforms those reported previously in the literature in terms of the TPR. Among all the existing schemes, the scheme that was proposed by Peking University (PKU) in 2016 generates the best performance [40]. Therefore, we are benchmarking our scheme against the one presented in [40].

Table 5.5 shows the performance comparison results of the proposed scheme in this thesis with the best existing scheme. The average bitrate is in Kilo-Byres per second (KBps). The three different bitrates shown are for three different cases, that are: selection based on blocks when the operating mode is 80 descriptors, selection based on displacements when the operating mode is 150 and selection based on displacements when the operating mode is 200 descriptors. Those results are all generated when the first approximation method is used in the persistent descriptors extraction stage.

As illustrated in Table 5.5, in all different bitrate budgets, our scheme uses lower bitrate but achieves better pairwise-matching performance. For example, when the bitrate is 4.95 KBps, the scheme of PKU achieves a TPR of 73.9%. In our scheme,

Table 5.5: Performance comparison of the proposed scheme with the best existing scheme

The scheme of PKU		Our scheme	
Bitrate (KBps)	TPR	Bitrate (KBps)	TPR
4.95	73.9%	4.79	82.22%
13.5	75.7%	9	86.67%
45.59	79.8%	12	87.22%

we decrease the bitrate to 4.79 KBps, and it achieves a TPR of 82.22%.

5.5 Conclusion

Three different methods for persistent descriptors extraction are proposed in this thesis. For the direct method, although it is the most accurate method, due to uncontrollability of the size of the persistent set, we cannot heavily rely on it. The computational complexity is another drawback, which makes it unpractical for real-world applications.

For the first approximation method, simulation results indicate that it performs well in providing a compact and descriptive representation for the GOP. When it is used to approximate persistent descriptors, for the operating mode that is 80, the block-based selection is the best. As the number of the selected descriptors increases, these three selection methods tend to have the similar performance. The best schemes under different operating modes have been shown to achieve excellent performance in video pairwise-matching.

For approximation method, one of the most significant advantages is that it utilizes motion information of videos to reduce complexity. However, when it is used to approximate persistent descriptors in the GOP, it provides poor performance under all operating modes.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this thesis, we have addressed the problem of descriptor selection from video sequences for the purpose of video pairwise-matching. In order to make use of temporal information of video sequences, the novel concept of descriptor persistency has been proposed. The set of extracted persistent descriptors is a good representation of the given video segment. The main focus of this thesis was the extraction and selection of persistent descriptors from video sequences.

An encoder has been presented in order to exploit the descriptor persistency. There are five main components in the designed encoder: keyframe labelling, SIFT descriptor extraction, persistent descriptors extraction, persistent descriptor selection and compression. Keyframes are used to break down the video sequence to GOPs in the first stage. Three different extraction methods have been proposed in this thesis: The first is the direct method, which extracts descriptors from all frames in the GOP and matches them. In order to reduce computational complexity, the second method extracts approximate persistent descriptors by matching two keyframes using ratio test and geometric consistency check in each GOP. Instead of applying geometric consistency check, the third method utilizes motion information using block-motion estimation, the goal is to remove the outliers from the approximate set. The persistent descriptor selection, which is the fourth stage of the encoder, adaptively control the size of the persistent descriptor set, where three different selection methods based on displacements, clusters and blocks, respectively, have been proposed. Also, a simple but efficient method for pairwise-matching has been proposed in this thesis.

Experiments have been carried out to evaluate the performance of the proposed schemes. The results show that when the operating mode is 80 descriptors, the combination of the first approximation and the selection based on blocks outperforms the other schemes. For the other operating modes, three selection methods tends to have the same performance.

6.2 Future Work

As we can see, extracting persistent descriptors is a good method for describing salient information in video sequences. However, there are still some aspects that can be explored:

1. In this thesis, persistent descriptors are approximated and selected in GOPs separately. However, whether there is correlation between two adjacent persistent sets or not is an interesting topic to be considered.
2. In the thesis, most of the selection approaches are based on the location part of descriptors. How to make use of the feature part of descriptors to generate

better selection method needs to be explored.

3. The computational complexity in the direct method stems from the need for adaptive GOP size. Thus, finding a method to generate adaptive GOP sizes is also interesting to consider.

Bibliography

[1] ISO/IEC JTC1/SC29/WG11/N15339, "Call for proposals for compact descriptors for video analysis (cdva) search and retrieval," Jul. 2015.

[2] ISO/IEC JTC1/SC29/WG11 N15338, "Evaluation framework for compact descriptors for video analysis - search and retrieval," Jul. 2015.

[3] G. Takacs, V. Chandrasekhar, N. Gelfand, Y. Xiong, W.-C. Chen, T. Bismpigiannis, R. Grzeszczuk, K. Pulli, and B. Girod, "Outdoors augmented reality on mobile phone using loxel-based visual feature organization," in *Proceedings of the 1st ACM International Conference on Multimedia Information Retrieval*, ser. MIR '08. ACM, 2008, pp. 427–434.

[4] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154–171, 2013.

[5] G. Takacs, V. Chandrasekhar, N. Gelfand, Y. Xiong, W.-C. Chen, T. Bismpigiannis, R. Grzeszczuk, K. Pulli, and B. Girod, "Outdoors augmented reality on mobile phone using loxel-based visual feature organization," in *Proceedings of the*

1st ACM international conference on Multimedia information retrieval. ACM, Oct. 2008, pp. 427–434.

[6] B. K. Horn, "The binford-horn line-finder," Dec. 1973.

[7] H. P. Moravec, "Obstacle avoidance and navigation in the real world by a seeing robot rover," DTIC Document, Tech. Rep., Sep. 1980.

[8] C. Harris and M. Stephens, "A combined corner and edge detector," in *Alvey vision conference*, vol. 15, no. 50. Citeseer, Aug. 1988, pp. 10–5244.

[9] C. Harris, "Geometry from visual motion," in *Active vision.* MIT press, 1993, pp. 263–284.

[10] Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong, "A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry," *Artificial intelligence*, vol. 78, no. 1, pp. 87–119, Oct. 1995.

[11] C. Schmid and R. Mohr, "Local grayvalue invariants for image retrieval," *IEEE transactions on pattern analysis and machine intelligence*, vol. 19, no. 5, pp. 530–535, May 1997.

[12] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[13] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, Oct 2005.

[14] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *European conference on computer vision.* Springer, 2006, pp. 404–417.

[15] V. Chandrasekhar, G. Takacs, D. Chen, S. Tsai, R. Grzeszczuk, and B. Girod, "Chog: Compressed histogram of gradients a low bit-rate feature descriptor," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2009, pp. 2504–2511.

[16] W. T. Freeman and M. Roth, "Orientation histograms for hand gesture recognition," in *International workshop on automatic face and gesture recognition*, vol. 12, 1995, pp. 296–301.

[17] F. Suard, A. Rakotomamonjy, A. Bensrhair, and A. Broggi, "Pedestrian detection using infrared images and histograms of oriented gradients," in *2006 IEEE Intelligent Vehicles Symposium*, Jun. 2006, pp. 206–212.

[18] ISO/IEC JTC1/SC29/WG11, "White paper on compact descriptors for visual search," Apr. 2013.

[19] H. Mansour, S. Rane, P. T. Boufounos, and A. Vetro, "Video querying via compact descriptors of visually salient objects," in *Image Processing (ICIP), 2014 IEEE International Conference on.* IEEE, 2014, pp. 2789–2793.

[20] A. Abbas, N. Deligiannis, and Y. Andreopoulos, "Vectors of locally aggregated centers for compact video representation," in *2015 IEEE International Conference on Multimedia and Expo (ICME)*, June 2015, pp. 1–6.

[21] Z. Huang, L. Y. Duan, J. Lin, S. Wang, S. Ma, and T. Huang, "An efficient coding framework for compact descriptors extracted from video sequence," in

2015 IEEE International Conference on Image Processing (ICIP), Sept 2015, pp. 3822–3826.

[22] M. Makar, V. Chandrasekhar, S. S. Tsai, D. Chen, and B. Girod, “Interframe coding of feature descriptors for mobile augmented reality,” *IEEE Transactions on Image Processing*, vol. 23, no. 8, pp. 3352–3367, Aug 2014.

[23] S. Jeannin and A. Divakaran, “Mpeg-7 visual motion descriptors,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 6, pp. 720–724, Jun. 2001.

[24] D. Tian, H. Sun, and A. Vetro, “Keypoint trajectory coding on compact descriptor for video analysis,” in *2016 IEEE International Conference on Image Processing (ICIP)*, Sept 2016, pp. 171–175.

[25] T. Lindeberg, “Discrete scale-space theory and the scale-space primal sketch,” Ph.D. dissertation, 1991.

[26] ——, “Scale-space theory: A basic tool for analysing structures at different scales,” *Journal of Applied Statistics*, vol. 21, pp. 224–270, 1994.

[27] D. G. Lowe, “Object recognition from local scale-invariant features,” in *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, vol. 2. Ieee, Sep. 1999, pp. 1150–1157.

[28] ——, *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[29] M. A. Fischler and R. C. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.

[30] O. Chum and J. Matas, “Matching with prosac - progressive sample consensus,” in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, vol. 1, June 2005, pp. 220–226 vol. 1.

[31] ISO/IEC JTC1/SC29/WG11/N14961, “Test model 12: Compact descriptors for visual search,” Oct. 2014.

[32] S. Lepsøy, G. Francini, G. Cordara, and P. P. B. de Gusmao, “Statistical modelling of outliers for fast visual search,” in *2011 IEEE International Conference on Multimedia and Expo*, Jul. 2011, pp. 1–6.

[33] J. J. Gibson, “The perception of the visual world.” 1950.

[34] B. D. Lucas, T. Kanade *et al.*, “An iterative image registration technique with an application to stereo vision,” 1981.

[35] B. K. Horn and B. G. Schunck, “Determining optical flow,” *Artificial Intelligence*, vol. 17, no. 1, pp. 185 – 203, 1981.

[36] G. Farnebäck, *Two-Frame Motion Estimation Based on Polynomial Expansion.* Springer Berlin Heidelberg, 2003, pp. 363–370.

[37] I. E. Richardson, *H. 264 and MPEG-4 video compression: video coding for next-generation multimedia.* John Wiley & Sons, 2004.

[38] Y. Q. L. L. J. C. D. H. Y. W. Muhammad Alrabeiah, Ting Yin, "Persistent descriptors: A new descriptor copression approach for video analysis," in *2017 IEEE International Conference on Image Processing (ICIP)*, Sept 2017.

[39] D. Arthur and S. Vassilvitskii, "K-means++: The advantages of careful seeding," in *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, ser. SODA '07. Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035.

[40] ISO/IEC JTC1/SC29/WG11, "Pkus response to mpeg cfp for compact descriptor for visual analysis," Feb. 2016.