CONTENT-BASED VIDEO COPY DETECTION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

SAVAŞ ÖZKAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

JUNE 2014

Approval of the thesis:

CONTENT-BASED VIDEO COPY DETECTION

submitted by **SAVAŞ ÖZKAN** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen _____

Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Gönül Turhan Sayan _____

Head of Department, **Electrical and Electronics Engineering**

Prof. Dr. Gözde Bozdağı Akar _____

Supervisor, **Electrical and Electronics Engineering Dept., METU**

Examining Committee Members:

Prof. Dr. A. Aydın Alatan _____

Electrical and Electronics Engineering Dept., METU

Prof. Dr. Gözde Bozdağı Akar _____

Electrical and Electronics Engineering Dept., METU

Prof. Dr. Uğur Halıcı _____

Electrical and Electronics Engineering Dept., METU

Prof. Dr. Adnan Yazıcı _____

Computer Engineering Dept., METU

Dr. Ersin Esen _____

Head of Image Processing Group, TÜBİTAK UZAY.

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Savaş ÖZKAN

Signature :

ABSTRACT

CONTENT-BASED VIDEO COPY DETECTION

Özkan, Savaş
M.Sc., Department of Electrical and Electronics Engineering
Supervisor: Prof. Dr. Gözde Bozdağı Akar

June 2014, 88 pages

In recent years, need in automatic video copy detection has been increased rapidly with the recent technical developments. In general, a developed system should provide a few requirements to conduct over large database including high detection accuracy, low comparison time and low memory usage. For that purpose, within the scope of the thesis, we propose a content-based video copy detection system that consists of three crucial stages namely feature extraction, quantization-based indexing and geometric verification. In feature extraction stage, local spatial and spatio-temporal features are extracted from reference and query videos to be used for similarity score calculation. In spatial domain, Scale Invariant Feature Transform (SIFT), Opponent SIFT, Flip Invariant SIFT (F-SIFT) and Speed Up Robust Transform (SURF) descriptors, in spatio-temporal domain, Histogram of Orientated Gradient (HoG) and Motion Boundary Histogram (MBH) descriptors are utilized. In the second stage, in order to make efficient comparison among local features, the local features are quantized into indices with three state-of-the-art indexing schemes Bag-of-word, Hamming Embedding and Product Quantization. In the final stage, since there would be outliers during matching content indices, a geometric post-processing stage is utilized for both spatial and spatio-temporal features that impose an overall geometric model to refine the accuracy. Additionally, a compact geometric signature that encodes the local relation of interest points with binary signature is computed. The experimental results are presented on the well-known TRECVID 2009 content-based video copy detection dataset. The experiments show that combination of Flip Invariant SIFT, Hamming embedding, enhanced weak geometric consistency and visual group binary signature yields the best overall result.

Keywords: Content-based Video Copy Detection, Near-Duplicate Video Search, Local Spatial Descriptors, Local Spatio-Temporal Descriptors, Quantization-based Indexing, Geometric Consistency, Visual Group Binary Signature.

ÖZ

İÇERİK TABANLI VİDEO KOPYA BULMA

Özkan, Savaş
Yüksek Lisans, Elektrik Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Gözde Bozdağı Akar

Haziran 2014, 88 sayfa

Son yıllarda, otomatik video kopya bulmaya olan ihtiyaç yeni teknolojik gelişmelerle birlikte artmıştır. Genellikle, geliştirilen sistemin büyük veritabanlarına uygulanabilmesi için birkaç zorunlu gereksinimi karşılaması gerekmektedir. Gereksinimler yüksek doğruluk bulma, düşük karşılaştırma karmaşıklığı ve düşük hafıza ihtiyacıdır. Bu amaçla, bu tez kapsamında, öznitelik çıkarma, nicemleme tabanlı indeksleme ve geometric doğrulama olmak üzere üç ana bölümden oluşan bir içerik tabanlı kopya bulma sistemi önerilmektedir. Öznitelik çıkarma bölümünde, referans ve sorgu videolardan, benzerlik hesaplamalarında kullanılmak için yerel uzamsal ve uzamsal-zamansal öznitelikler çıkartılmaktadır. Uzamsal uzayda, ölçek değişimsiz öznitelik dönüşümü (SIFT), Karşıt SIFT, Çevirme Değişimsiz SIFT ve Hızlandırılmış Dayanıklı Dönüşüm (SURF) tanımlayıcıları, uzamsal-zamansal uzayda, Dönüşüm Değişimleri Histogramı (HoG) ve Hareket Sınır Histogramı (MBH) tanımlayıcıları hesaplanmıştır. İkinci bölümde, yerel öznitelikler arasında hızlı karşılaştırma yapabilmek için, yerel tanımlayıcılar, üç yeni önerilen Kelime Çantası, Hamming Yerleştirme ve Çarpım Nicemlemesi indekleme metodlarıyla indekslere nicemlenmektedir. Son adımda, öznitelik indekslerinin eşleşmesi sırasında doğru eşleşmeyen noktalar olabileceği için, her iki uzamsal ve uzamsal-zamansal öznitelikler için, geometrik modeli uygularak sonuçları iyileştiren bir geometri son basamağından faydalanılmaktadır. Ayrıca, yerel ilgi noktaları bilgisini kodlayan

yoğun bir geometrik imza hesaplanmaktadır. Test sonuçları, iyi bilinen TRECVID 2009 içerik tabanlı kopya bulma veritabanında sunulmaktadır. Sonuçlar göstermektedirki, Çevirme değişimsiz SIFT, Hamming yerleştirme, iyileştirilmiş zayıf geometrik tutarlılık ve görsel grup ikili imza kombinasyonu en iyi toplam sonucu vermektedir.

Anahtar Kelimeler: İçerik Tabanlı Video Kopya Bulma, Benzer-Çift Video Arama, Uzamsal Yerel Tanımlayıcılar, Uzamsal-Zamansal Yerel Tanımlayıcılar, Nicemleme Tabanlı İndeksleme, Geometric Tutarlılık, Görsel Grup İkili İmza

To my family...

ACKNOWLEDGEMENTS

First and foremost I would like to express my gratitude and appreciation to my dear supervisor Prof. Dr. Gözde Bozdağı Akar who had always presented her endless help and guidance during this study even I lost my motivation sometimes. Probably, I could never finish this thesis, if she did not support and guide me.

I would like to thank to Dr. Ersin Esen who is one of the wisest people that I have ever met in my lifetime. I am grateful to him for the priceless discussions that we had made in his lodge, useful comments, giving me a flexibility to explore my own ideas and many more. Shortly, he is the person who deserves the phrase “Ersin Abi” in real terms.

I am thankful to Prof. Dr. A. Aydın Alatan who introduced the greatness of computer vision and machine learning fields and extended my research vision.

I have pressure to work with two wonderful people in my workspace as colleagues, Dr. Medeni Soysal and Dr. Engin Tola who motivated me to improve my writing skill with their satires in polite ways.

I would like to thank *iTunes Radio*® and all alternative music bands namely *Pera*, *Rise Against*, *Safetysuit*, *We Are The Ocean* and many more that kept my motivation high during code implementation and test.

Finally, I would be nothing without their unconditioned love and supports, my mother Serap, my father Arif and my brother Akın. It is good to know that they will be always with me whenever I need a help.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURES

LIST OF TABLES

TABLES

CHAPTER 1

INTRODUCTION

1.1 Introduction

With the developments in internet technologies and proliferation of multimedia sharing websites, unprecedented copyright infringements have emerged in the recent years. Due to the fact that amount of circulated multimedia data over internet reach to huge vast, handling this data with bare human-based interactions becomes impractical. Thus, demands for finding a generic solution have had immense attractions than ever before. By the help of the advancement on computer vision and machine learning, researchers have been investigating this problem under the title of copy detection [1-19].

The primary goal of copy detection is to search a query video in a large reference video archive and obtain if original source of the video is in this reference archive or not without any external intervention.

Frequently, in order to impede identification of source data, several deformations are inserted purposely into original data source including compression, scaling, cropping, picture-in-picture, insertion of text, etc. [58].

As inferred from its usability, this field can be used as a base in many applications like content-based search [20], advertisement tracking [21] and multimedia linking [22] successfully.

In literature [1-19], the studies have gathered on two complimentary approaches, digital watermarking [19] and content-based copy detection [1-18]. The core idea of digital watermarking is to embed irreversible signatures into media which can be either visible (text

or logo) or invisible (it would not be perceived by human-eyes) for future copy determination. A deficiency of this method is that as the embedded signatures are sensitive to geometric transformation and compression, it needs a pre-modeling stage to model these signatures to all possible attacks in advance. Inherently, this causes an increase on storage size of the data alongside of itself. Since these signatures should be inserted before video is released; the method cannot be deployed on currently circulated data over internet.

Second approach is content-based copy detection. The underlying assumption is that instead of embedding any information; content signatures are extracted from multimedia data. Thus, this procedure makes this approach easier and applicable to copy detection without increasing video size or needing any pre-modeling stage.

Mainly, this approach consists of two main steps. In first step, known as *offline step*, sufficiently distinctive signatures are obtained from multimedia contents using different feature models and different sampling strategies, and an archive is constructed by storing these signatures for future copy search. In second step, which is called *online step*, query signatures are compared within all the reference database to determine whether query data is transformed from this reference archive or not.

Typically, such an automatic copy detection system should provide couple of essential requirements when large amount of the data is considered. Two properties come in prominence alongside of high detection accuracy: low computational complexity and less burden of memory usage. In the scope of this thesis, we have explored these three essential aspects via configuring different computer vision and machine learning techniques.

1.2 Related Works

As we have mentioned in section 1.1, content-based copy detection can be distinguished into two crucial steps as *offline* and *online*. In offline step, by exploiting visual, temporal and/or even aural contents of video, sufficiently discriminative and robust signatures are extracted.

Saraçoğlu et al [1] assert that audio content would appear to be more robust over utilizing visual and temporal contents. Interestingly, from their experiments, they observe that joint usage of visual and audio contents makes significant improvements on detection rate. Nevertheless, studies in literature [2-18] have predominantly concentrated on visual and temporal contents owing to frequency of attack incidences.

Feature extraction methods can be grouped as local and global according to their representation procedures [23-25, 39]. Global signatures represent color, edge, texture and motion properties with single feature vector that is obtained from whether an entire frame or concatenation of sub-partitioned frame windows [39, 41].

In [2], spatial average intensity variations are modeled on 2×2 sub-partitioned successive frames. According to the author's statements, this provides extra robustness particularly on intensity and brightness changes. In [3], authors explain that effective copy detection should be robust against changes in spatial and temporal variations with low calculation cost. Thus, they accept global signature matching stage as a probabilistic model and turn into a graph problem. As authors emphasize that this approach yields fair results on the attacks which aim video sequence like frame dropping. In another study [4] that is proposed by again this team, statistical characteristics of pair wise correlation of sub-partitioned frames are computed by employing average intensity on each partition. Besides fast computational capability, according to author's explanation, this representation is robust to signal-based attacks including contrast and blurring changes.

At the side of motion content, Taşdemir et al [5] assert that motion is useless under high sampling rate and according to their evaluation results, applying lower sampling rate (for example, 5 frame per second) gives promisingly better accuracy with distributions of magnitude and angle of motion features. Using the previous contribution, Roopalashmi [6] investigates this task with couples of novel motion features including motion intensity (mean and deviation of motion magnitude), spatial distribution of activity (predicts the active

regions) and dominant direction of activity (dominant direction of motion) which all extracted globally.

In [7], in addition to low-level global visual signatures and sequence matching, facial human appearance features are utilized. According to their statements, interestingly, lower side of human body appears more distinctive than face region due to clothing color and background for duplicate video detection.

However the weak spot of modeling contents with similar manner is that owing to lack of invariance against scaling and cropping, global feature extraction methods would fail on picture-in-picture and cropping attacks which contain geometric transformations. Although dividing frame into sub-windows inserts distinct localization information, it would still deteriorate accuracy for geometric attacks.

In local signature [23-29], video content is represented around sparsely or densely sampled points that are scale and rotation invariant [23, 25]. Inherently, this strategy gives strength to occlusion clutter alongside of geometric transformation because of the ability to make comparisons on these features individually.

However the main disadvantage of local signatures is that total number of local patches is prohibitively high to represent the video completely. Hence, this causes a deceleration on comparison stage. Particularly, considering large amount of media collection, direct use of these signatures becomes senseless. Although dimension reduction that changes the feature space with smaller one seems as a primitive solution for scene understanding [41] and object detection [43], it might lead an overshooting when noisy version of signal is encountered.

Although dimension of signature become smaller, comparison stage would still take place exhaustively and it creates a redundancy on comparison stage. Hence, in literature [45, 48, 49], quantization-based indexing procedure are generally deployed due to effectiveness on memory usage and efficient search capability. The simple idea relies on mapping local

features into indices by finding corresponding cluster center from pre-clustered feature space. Expectedly, during the quantization stage, there would have some information losses on distinctive power of feature vector.

To be able to conduct an effective search on a large dataset, the storing procedure should be integrated with inverted index data structure [45]. This structure is composed of descriptor entries where each are associated with indices. Therefore query descriptor is only compared with the same indexed descriptor and this greatly reduces the search complexity.

Ateş et al [8] present a case study that investigates the performances of SURF [28] and SIFT [25] local visual features on this task. Their results show that the performances of two local features are nearly equal. Also, for smaller codeword size like from 128 to 1024, the increment of codebook size enhances performance alongside of acceleration of comparison speed. In [9], authors emphasize that predominant drawback of derivation-based descriptors [25] is that derivations are only taken in x and y directions. For that purpose, they propose a novel visual feature extraction scheme that prevents underestimation of other direction. This representation provides robustness particularly on the attacks that aim the signal content like compression. Heritier et al [10] concentrate on efficient signature indexing and thus they propose hierarchical indexing mechanism.

In the recent years, to describe the regions more precisely, combining local spatial and temporal descriptors as single feature has been appeared as a hot-topic. Extensively, this representation is deployed on action recognition [35, 36] and it is known as *spatio-temporal* or *trajectory-based* representation. The core advantage of this type of feature model is that spatial content variations on consecutive frames are exploited in feature extraction. Thus, it helps to augment the distinctivity of feature. Law-To et al [11] assert that employing visual content singly cannot model the sequential variation which might occur while small transformation. Thus, they propose the method that combines visual and temporal contents. The idea is to extract local spatial features on successfully tracked consecutive frames in time.

Similarly, in [12], a novel trajectory-based signature is introduced that encodes the relative spatial position of each tracked point in proceeding time instance instead of any visual content.

The crucial observation of Wilems et al [13] is that local visual features need to be computed on uniformly sampled frames. Hence, in spatial domain, matching stage takes a few frames into account. As the nature of spatio-temporal features, interest points are computed on spatial and temporal spaces jointly. Therefore this method extends the matching stage from a few frames to entire video.

Even if selection of best feature and indexing schemes seem as two core steps for successful copy detection, in literature, there are several crucial observations and solutions that might improve accuracy of detection even better. In [14], to increase the accuracy on strong encoding and picture-in-picture attacks, all the reference videos are modeled with these attacks in advance with various configuration parameters and they are stored alongside of raw visual content signatures. According to their results, proposed method improves the performance particularly on the pre-modeled attacks. However it causes notable increases on memory usage and computational complexity. In other work [15], authors assert that logos, banners and texts generate many mismatching results owing to the occurrence frequency and repeatability. Hence, they propose an algorithm that detects the text, logo and banner on frame. Later, it discards the local signatures that overlap with these regions for future copy determination. Uchide et al [16] state that since global features are not scale invariant, they do not work well when frame undergoes change in scale. To mitigate this drawback, they propose a picture-in-picture boundary detection algorithm for query video that simply accumulates image gradient in x and y directions.

As stated previously, the deficiency of quantization-based indexing on local features is that geometric relations among local signatures are discarded while comparing similarity of indices. For that purpose, Douze et al [17] appends two post-filtering stages to refine true corresponding matches in spatial and temporal domains. In this system, weak geometric consistency [48] method and 1D hough estimation techniques are deployed in spatial and

temporal domains respectively. The evaluation results validate that these simple but effective post-processing stages make a drastic contribution on performance. In another study [14], in order to maintain the trade-off between scalability and robustness, authors combine local visual signatures with hashing-based indexing and 2D homography [50] estimation.

In a comparative work, which is done by Law-to et al [18], the performances of several state-of-the-art local and global descriptors are analyzed comprehensively. The first observation is that even if local visual features have excessive computation cost rather than global visual features, their performances seem as optimum. The second observation is spatio-temporal features work well in small transformations.

1.3 Scope of Thesis

As we stated in section 1.1, convenient copy detection should provide couple of core requirements. High success rate can be accepted as an indispensable necessity among others. Therefore, selecting discriminative and robust features would have a crucial importance for future copy determination. The recent studies [23, 25] validates that even if global signatures yield fast and compatible results on this task, they lack invariance on a few attacks that include geometric distorters. Hence, local feature extraction methods are frequently exploited owing to the robustness to the attacks. Also, this type of representation has invariance to illumination and compression [25].

Although local signatures yield complementary results on geometric attacks, large amount of features need to be computed from single image to detect the duplicate frame pairs truly. As expected, this large amount triggers undesirable increases on memory usage and slows down calculation speed.

Hence, dimension reduction should be applied over feature vectors in order to make prompt comparisons. In literature [45, 48, 49], there are couple of approaches to mitigate this bottleneck of local descriptors. Utilization of quantization-based indexing schemes have

shown superior performances on comparison speed and accuracy. Thus, in the recent years, combination of this type of approach with inverted index data structure [45] have been introduced on many computer vision tasks seamlessly. The core idea is to map feature vectors to indices or more correctly for this domain *visual words* [45] using a pre-clustered feature space.

Although the joint usage of local descriptors with quantization-based indexing schemes yield excellent results, these representations discards the geometric relation that exists among local signatures. Thus, use of the geometric relations enables to improve the performance even further. Even though there are couple of state-of-the-art homography [50, 51] estimation and local neighboring methods [55-57], their comparison complexities limit their applicability on entire archive. Therefore, instead of applying these kinds of obstructive estimations, seeking a simpler geometric verification stage by investigating characteristic of geometric parameter distribution of local signatures and exploring local geometric relation with compact signature would be more truly suitable.

For all these reasons, in this study, video copy detection task is revisited by employing content-based approach. First, local spatial and spatio-temporal features [25-29, 33] of video are exploited. Then, in order to make efficient search, these features are represented with quantized-based indexing signatures [45, 48, 49] and the similarity scores for all frames are calculated using these signatures. In the final stage, geometric consistencies [48, 54] among corresponding local signatures are investigated to enhance the accuracy by introducing negligible amount of increase in comparison complexity and memory.

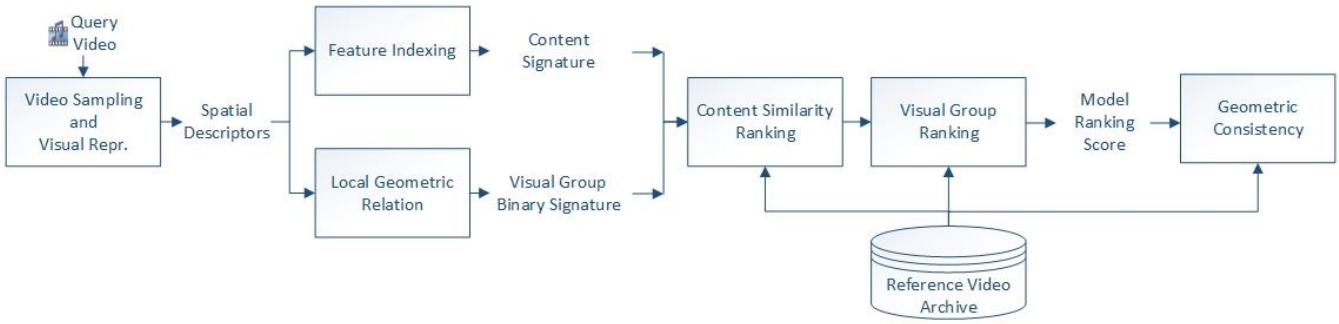

Figure 1. Overall block diagram of the developed copy detection system for spatial domain.

1.4 Outline of Thesis

As we stated in previous sections, successful content-based copy detection should consist of three cascaded stages namely feature extraction, indexing and geometric verification. Therefore we have partitioned this thesis into three main chapters where each stage is explained thoroughly.

Since the scope of the thesis bases on local features, necessary background on spatial and spatio-temporal feature extractions are summarized in separate sections in *Chapter 2*. To provide similar conditions for each feature model, each one is computed on uniformly sampled one second interval frames. Before giving the essential information about each feature method, the underlying procedures of interest point detection is investigated at the beginning of each spatial and spatio-temporal sections.

The core information about three existent quantization-based indexing methods are given in *Chapter 3* from simple to complicated. Additionally, we propose unique soft-assignment similarity score metrics for all indexing methods. Data structure which provides an effective signature search capability and weighting scheme according to their term-frequencies are discussed in detail in this chapter.

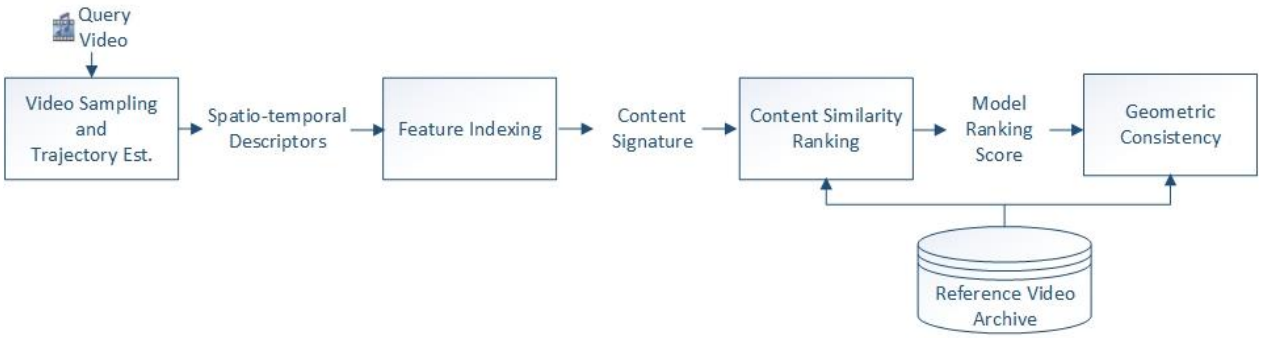

Figure 2. Overall block diagram of the developed copy detection system for spatio-temporal domain.

Chapter 4 is devoted to establish fast and compact geometric verification among local signatures. To improve the understandability of the geometric verification stage, this chapter is divided into two sections namely geometric consistency and local geometric signature. Even though there are several methods that aim to constitute a weak geometric consistency in spatial domain, in the scope of the thesis, we propose a novel trajectory-based geometric consistency for spatio-temporal signatures which gives superior performance. Additionally, we have reintroduced the weak geometric consistency for spatial domain to gain invariance against flip transformation. In local geometric signature section, we propose a novel local geometric signature that simply encodes the local interest point relation in neighboring area as a single compact binary signature. This signature helps to discard the outliers that are obtained from content similarity search with small burden of memory and comparison.

In *Chapter 5*, according to obtained evaluation results on a dedicated dataset [56], positive and negative aspects of the combination of each feature extraction, indexing and geometric verification schemes are discussed around three essential properties which we give in *Motivation* section. The overall block diagrams for spatial and spatio-temporal domains are shown in Figure 1 and Figure 2, respectively.

Finally, in *Chapter 6*, the conclusion of the thesis is presented and future research direction is discussed according to weakness of the proposed methods.

CHAPTER 2

FEATURE EXTRACTION

In image processing fields, digital image consists of pixel numbers that correspond to integer or floating points. Even though these pixel numbers yield discriminative information about content of an image, most of them are redundant. Thus, it can decrease the performance of detection/classification significantly. Similar to image, video consists of successive images and this makes the problem even harder.

Therefore, image/video content should be represented as a set of features that can vary according to desired application or task. In general, the assumption is that the content is depicted by obtaining distinctive characteristic information about pixel distribution. For example, while color, texture and gradient features give superior performance on many spatial domain applications [25-27, 32, 39], trajectory and motion features are preferred in temporal domain [33, 39].

Selection of feature model can be introduced according to problem specifications. The primitive approaches in literature are based on converting full size of pixel values into lower dimension by using dimension reduction methods [37, 38]. They use the observations that are learned in advance to decide pixel relation is important or not. Although this scheme yields impressive performance on character recognition [38] and face detection [37], they lack robustness and invariance on multimedia content representation. Hence, utilization of color, edge or texture would give more accurate results especially on that task [25-27, 39].

In literature, feature extraction methods can be distinguished into two modes according to way of representation of multimedia content namely global [39] and local [25] extractions. In global feature extraction, the contents of multimedia data that can be either image or video,

are represented with a single feature vector that describe the whole data by representing its color, edge texture or motion information.

To insert localization information into the representation for enhancing the distinctive power, multimedia data is partitioned into sub-regions in spatial and temporal domains. Then, all feature vectors from each region are concatenated into a single vector.

The main advantage of extracting global features is that they give instant response to calculation and comparison stages besides fair results. Particularly on semantic scene understanding [40], this type of features gives quite good results.

Although these features yield fair results on multimedia data, inherently, they are sensitive to geometric transformation and occlusion clutter because of describing the data with single feature vector. As expected, these feature vectors are partially robust in a range of distortion and it would induce an ambiguity in comparison stage.

In the recent years, owing to their distinctive power and robustness to occlusion and geometric transformation, local feature extraction methods [23-27] which are extracted around interest points, have been applied on nearly all applications in computer vision task including image retrieval [42, 49], video data mining [45], object recognition and localization [43], scene understanding [41] and camera calibration [44].

The underlying assumption is to detect local patches on an image which are invariant to geometric transformations and describe the content inside these patches. Similar to global features, color, edge, texture or motion content is exploited.

Frequently, local feature extraction consists of two steps. First, interest points that are robust to geometric transformations and thereby view point changes are determined [25]. Second, leveraging location and scale parameters of these points, a circular region is defined around

each point and a feature vector is computed by exploiting distinctive content information [25-27].

Of course, local descriptors have several deficiencies that are mostly encountered on calculation and comparison stages. Since feature extraction and signature estimation are done in offline step of content-based video copy detection task, there is no time limitation to construct the reference archive.

For all these reasons that we have mentioned above, local descriptor can give better performances alongside of high computation limitations. In this work, we have utilized several local feature extraction schemes in spatial and spatio-temporal domains. Hence, this chapter will be partitioned into two sections and the detailed information about each type of extractions will be presented. First, for spatial domain, an existent approach for interest point detection will be summarized. Additionally, four local region-based visual descriptor methods will be explained thoroughly. In second section, similar to first section, the interest point estimation and representation will be presented for spatio-temporal domain.

2.1 Local Spatial Feature Extraction

The higher reliability score between two images is directly related with finding more true correspondences on the images. Hence, the detection of robust interest points and describing these region around interest points have crucial influences on true image matching. For that purpose, to improve the clarity, we will explain interest point detection and describing the regions in two separate sub-sections.

2.1.1 Interest Point Detection

In literature [23-25], there are several interest point detectors that follow similar assumptions. The common assumption is that such a point should be stable to translation, scale and orientation changes and robust enough to quality decrease. Hence, the fact is that interest

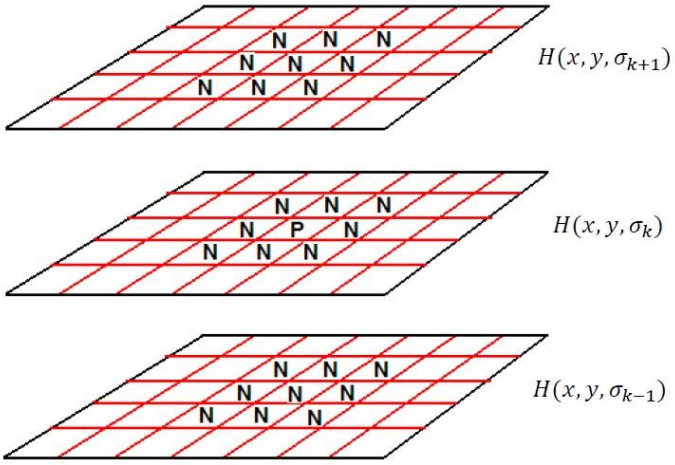

Figure 3. Scale-space Hessian Matrixes. P denotes the central point and N presents the neighbouring points around point P.

points should be located on either blobs or edge corners. This provides extra strength on view point changes and decrease on quality.

In this work, firstly, we will utilize a state-of-the-art interest point detector which is called as Hessian-Laplacian [23, 24]. This methodology detects interest points on blobs with a scale parameter. Hence, in order to obtain location of interest point and characteristic scale, Hessian matrix and Laplacian function are computed respectively.

Hessian matrix consists of second order partial derivatives that is derived from Taylor series expansion. This matrix is frequently utilized on analysis of the local image structure. This matrix measures the curvature at a point using neighboring intensity values. The assumption is that the eigenvectors of matrix yield the maximum and minimum curvature directions for a point. Additionally, the eigenvalues give the magnitude of curvatures on eigenvector's directions. Hessian matrix can be written as:

$$H(x, y) = \begin{bmatrix} I_{xx}(x, y) & I_{xy}(x, y) \\ I_{yx}(x, y) & I_{yy}(x, y) \end{bmatrix} \quad (1)$$

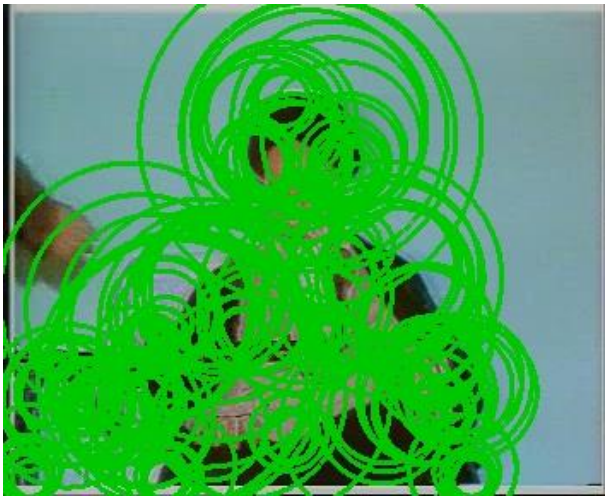

Figure 4. Local Hessian-Laplacian interest points. Green circles defines the regions.

where I_{xx}, I_{xy}, I_{yx} and I_{yy} second order derivatives in specified directions.

Therefore, the determination of Hessian matrix would help to obtain the point with strong spatial variations. Leveraging this property of Hessian matrix, first, a scale-space is constructed by convolving the input image with various of Gaussian kernels $G(x, y, \sigma_k)$ where σ_k is the variance of k level of Gaussian kernel as:

$$L(x, y, \sigma_k) = I(x, y) * G(x, y, \sigma_k) \tag{2}$$

Since Hessian-Laplacian method is a blob like detector, the purpose of convolving image with Gaussian kernels is to smooth the signal and obtain the initial candidate value of scale parameter for this point. For each scale-space convolved images, Hessian matrix is computed as:

$$H(x, y, \sigma_k) = \begin{bmatrix} L_{xx}(x, y, \sigma_k) & L_{xy}(x, y, \sigma_k) \\ L_{yx}(x, y, \sigma_k) & L_{yy}(x, y, \sigma_k) \end{bmatrix} \tag{3}$$

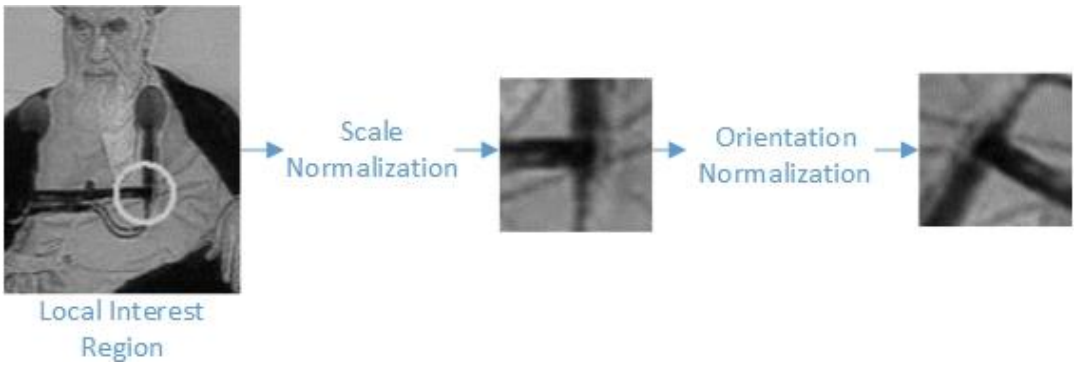

Figure 5. The block diagram of interest region normalization.

To detect the stable interest points, determination of Hessian matrix on single point is compared with 3×3 neighborhood area in spatial (8 points) and scale points (2x9) as shown in Figure 3. If the central point is of a greater value among neighboring points and a given threshold, it is selected as an interest point. The reason for using a threshold is to eliminate the points that have weak maxima.

After obtaining the location and the initial scale of the interest point, the scale value is refined in order to assign more proper characteristic. According to author's statement [23], Laplacian function is more suitable to determine the characteristic scale from an image structure [23]. Hence, Laplacian function is incorporated for different size of Gaussian kernels as:

$$Lap(x, y, \sigma_k) = {\sigma_k}^2 \left| L_{xx}(x, y, \sigma_k) + L_{yy}(x, y, \sigma_k) \right| \quad (4)$$

Similar to interest point detection with Hessian matrix, Laplacian function is computed over all scales. The scale which gives the maximum value among neighbors is selected as the characteristic scale.

2.1.2 Spatial Descriptors

With location, scale and orientation (assignment stage will be explained in descriptor methods) of interest point, we define circular regions around that point as illustrated in Figure 4. To provide same condition on the interest points that have different scale and orientation parameters, we utilize a transformation stage before describing the regions. The block diagram of transformation stage is given in Figure 5. First, the region around interest point with characteristic scale are normalized into 41×41 patch. Second, the region is rotated using orientation parameter around the interest point. In final stage, visual content is exploited over these normalized region.

In this work, for spatial domain, we have implemented four state-of-the-art visual feature extraction methods including Scale Invariant Feature Transform (SIFT) [25], illumination invariant color version Opponent SIFT [26], flip invariant version F-SIFT [27] and Speed-Up Robust Feature (SURF) [28].

In the following sub-sections, we will explain the details of these spatial descriptors.

2.1.2.1 Space Invariant Feature Transform (SIFT)

The original of the descriptor [25] consists of four major stages namely scale-space extrema detection, keypoint localization, orientation assignment and keypoint descriptor. First two stages that are also known as Difference of Gaussian (DoG) [25] and they correspond to interest point detection. Since, we have utilized Hessian-Laplacian for that purpose, in here, we will not give any information about these stages. Thus, we will jump directly to orientation assignment stage.

In orientation assignment stage, in order to preserve the robustness against orientation changes or image rotation, an orientation parameter is calculated based on local property of interest

point. In this method, first, inside the region of interest point, gradient magnitude $m(x, y)$ and gradient orientation $\theta(x, y)$ is computed on each pixel (x, y) using pixel differences as:

$$m(x,y) = \sqrt{\big(I(x+1,y) - I(x-1,y)\big)^2 + \big(I(x,y+1) - I(x,y-1)\big)^2} \tag{5}$$

$$\theta(x,y) = \tan^{-1}\left(\frac{I(x,y+1) - I(x,y-1)}{I(x+1,y) - I(x-1,y)}\right) \tag{6}$$

Then an orientation histogram is constructed. The size of histogram is selected as 36 where each bin covers 10 degree range of orientations. The point inside the region is added up according to quantized orientation value with gradient magnitude. To increase the importance of the point that are close to center of interest point, Gaussian weighting is applied over gradient magnitudes [25].

The underlying idea of estimation of orientation in local region is based obtaining peak value of accumulation of gradient orientations in a histogram. The peak bin corresponds to dominant direction of local gradients. However, because of the quantization of gradient orientation and noise of the image, selecting single dominant direction would be inaccurate. Therefore, author emphasizes that [25] local peaks that are up to %80 of the highest peak must be considered as orientation characteristics of interest point.

In keypoint descriptor stage, instead of utilizing grayscale intensity values directly, gradient magnitude and orientation distribution are employed. First, in order to insert the distinctive location information in feature vector, the region is partitioned into 4×4 sub-regions. Then, from each sub-region, an orientation histogram is computed with 8 directions and weighted with gradient magnitudes. At the end, the orientation histograms are concatenated and the final feature vector length would be equal to $4 \times 4 \times 8 = 128$.

2.1.2.2 Opponent SIFT

Scale Invariant Feature Transform (SIFT) is applied on grayscale image to represent the visual content of local region. In real-world scene, image consists of additional color channels. However, direct use of SIFT feature on RGB (Red, Green and Blue) or YUV (luminance, chrominance blue, chrominance red) induces an ambiguity due to the fact that changes in the illumination greatly affect the performance of matching or object recognition for these color spaces.

In [26], authors express a couple of illumination change conditions including light intensity/light color changes and shifts with linear equations. Leveraging the conclusions of these formulations, they propose a novel color space which consists of three color channels named as *Opponent Color Space.*

The thought is to convert red, green and blue values into new invariant color space by combining with each other. The combination of red, green and blue channels can be express as:

$$\begin{pmatrix} O_1 \\ O_2 \\ O_3 \end{pmatrix} = \begin{pmatrix} \frac{R-G}{\sqrt{2}} \\ \frac{R+G-2B}{\sqrt{6}} \\ \frac{R+G+B}{\sqrt{3}} \end{pmatrix} \tag{7}$$

where intensity information is represented with O_3 and color information by O_1 and O_2 channels.

In Opponent SIFT descriptor, orientation parameter of the region is computed on grayscale value similar to SIFT descriptor. Differently, all opponent color channels are accepted like grayscale and from each, 128 dimensional feature vector is computed using SIFT descriptor.

At the end, these three channel SIFT features are concatenated as one and $3 \times 128 = 384$ dimensional feature vector is obtained.

2.1.2.3 Flip-Invariant SIFT (F-SIFT)

Although SIFT descriptor is invariant to scale and orientation changes, it is not invariant to flip transformation in any axis. The source of this problem is based on that the insertion of locality in feature representation is not robust to this type of transformation even if interest point detectors are. In [27], a novel method is proposed that preserves the originality of SIFT descriptor extraction including grid-based structure alongside of enrichment on flip invariance.

The intuitive idea is to make the region invariant by transforming this patch over the direction where the flip incident has been occurred before visual feature extraction. For that purpose, there should be a rule that determines the flipping action should be performed or not.

Hence, authors [27] propose dominant curl computation for this problem. This computation defines a vector operation that describes the infinitesimal rotation of a vector field. The direction of curl corresponds to the axis of flip. In multivariate calculus, the curl of $\boldsymbol{F}$ where $F(x, y, z)$ is a vector field is given as:

$$\nabla \times F = \begin{vmatrix} i & j & k \\ \dfrac{\partial}{\partial x} & \dfrac{\partial}{\partial y} & \dfrac{\partial}{\partial z} \\ F_1 & F_2 & F_3 \end{vmatrix} \tag{8}$$

With Stokes' theorem, this equation turns into integration of curl as:

$$\iint_{\Sigma \in R^3} \nabla \times F \cdot d\Sigma \tag{9}$$

In spatial domain, curl computation is defined in 2D discrete vector field and it is equal to cross product of first order partial derivatives along x and y directions respectively as:

$$C = \sum_{(x,y)\in \boldsymbol{I}} \left(\sqrt{\frac{\partial I(x,y)^2}{\partial x} + \frac{\partial I(x,y)^2}{\partial y}} \right) \times \cos(\theta_r(x,y)) \tag{10}$$

where

$$\frac{\partial I(x,y)}{\partial x} = I(x-1,y) - I(x+1,y) \tag{11}$$

$$\frac{\partial (x,y)}{\partial y} = I(x,y-1) - I(x,y+1) \tag{12}$$

$$\theta(x,y) = \tan^{-1}\left(\frac{I(x,y-1) - I(x,y+1)}{I(x-1,y) - I(x+1,y)}\right) \tag{13}$$

$$\theta_r(x,y) = \theta(x,y) - \tan^{-1}\left(\frac{y}{x}\right) \tag{14}$$

C denotes the possible direction in clockwise or counter clockwise manners according to its sign. In this work, negative sign indicates the region should be flipped. Thus, before employing normalization onto the region according to dominant orientation as in Figure 5, for flip detected region, region is flipped in vertical axis and later orientation normalization is deployed. Similarly, for visual extraction, SIFT descriptor is computed over flip normalized region and 128 dimensional feature vector is computed.

2.1.2.4 Speed Up Robust Feature (SURF)

Speed-up robust feature (SURF) [28] propose an alternative to SIFT descriptor for object recognition and interest point matching. The main contribution of this descriptor according to author's statement is that it computes local features several time faster alongside of superior performance.

Similar to SIFT, it consists of interest point detection, orientation estimation and describing interest point steps. For orientation assignment, Haar-wavelet responses are calculated in x and y directions by combining integral image scheme. The dominant orientation is estimated by calculating the sum of all haar-wavelet responses within sliding orientation window. Similarly, the peak of orientation histogram gives the orientation characteristic of points.

For feature extraction, in order to preserve the spatial information, the region is divided into 4×4 sub-regions. From each sub-region, horizontal d_x and vertical d_y wavelet responses are summed up and a vector is formed as $v = \left(\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|\right)$ where $|d_x|$ and $|d_y|$ denote the absolute value of d_x and d_y respectively. Thus, the final dimension of feature vector is equal to $4 \times 4 \times 4 = 64$.

2.2 Local Spatio-Temporal Feature Extraction

Naturally, video consists of successive images and it contains temporal variations of frames in addition to visual content. Hence, combination of visual content with temporal variations would yield better distinctive representation. For example, for an object with same background gives similar spatial features for different scenes on videos. However, unique movements of the object in different scenes can create a discrepancy. Thus, utilizing temporal variation alongside of visual content can improve the accuracy.

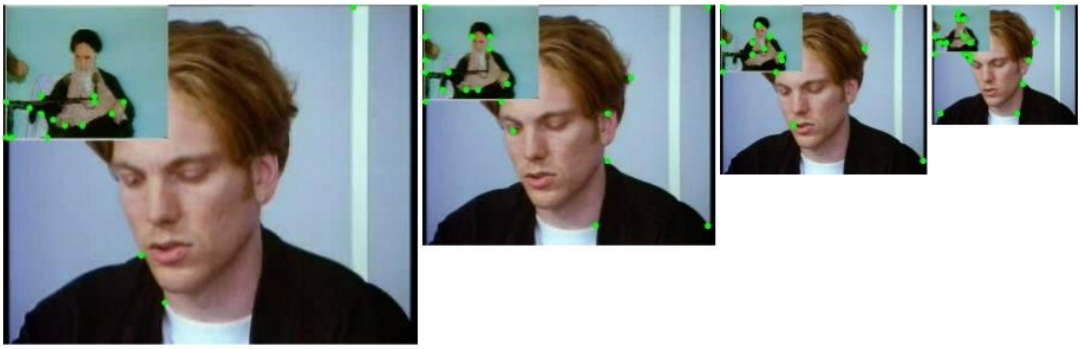

Figure 6. Local interest points at multiple scales.

Joint usage of temporal and spatial contents of video is extensively investigated on action recognition task [29, 35, 36]. In literature, two common approaches can be utilized for spatio-temporal feature extraction. First approach [35, 36] is that similar to spatial interest point detector, scale invariant regions are obtained on spatial and time domains jointly which is known as space-time interest points. Then, inside this space-time region, content of video is described.

Second approach is that interest points are detected on frames and tracked within specified time interval [29]. Thus, temporal variations of trajectory points are added into feature in addition to spatial content.

In this work, we have used trajectory-based spatio-temporal feature extraction scheme. Similarly, this method has two steps including interest point detection and tracking in consecutive frames and describing content of interest points.

2.2.1 Interest Point Detection and Tracking

We have utilized dense trajectory estimation method [29] to detect and track the interest points. According to author's statement, this representation captures the foreground motion information with high precision.

In the proposed methods, first, interest points are densely sampled on frames. In order to consider the scale changes, image pyramid that consists of downsampled version of input frame in scale, is constructed and sampling is also carried out on these images. Hence, this sampling strategy makes sure that interest points cover all spatial positions in different scales.

Since trajectory will be estimated on these points by tracking in time, these points should be sufficiently stable. Therefore, authors propose a filtering stage that eliminates the interest points which are on homogeneous areas. The assumption is that [30] if eigenvalues of auto correlation matrix of point is smaller than an adapted threshold, this point is discarded. Empirically, authors set the threshold value as:

$$T = 0.001 \times \max\left(\lambda_i^1, \lambda_i^2\right) \tag{15}$$

where λ_i^1 and λ_i^2 are the eigenvalue of a point i on the image. An example for densely sampled interest points on spatial scales are shown in Figure 6.

Before given the detail about the interest point tracking, this method tracks the interest points on each spatial scale separately. This means that transitions between different scale points are ignored.

In tracking stage, first, on each spatial scale, optical flow field [31] is constructed. Additionally, in order to reduce the sensitivity of motion field, 3×3 median filter is applied around all interest points.

In this work, we have used the motion content of video, static trajectory in other word the trajectory with small variations or large displacements has been filtered out.

2.2.2 Spatio-Temporal Descriptor

After tracking the interest points in consecutive frames successfully, spatio-temporal feature is calculated in space-time volume whose size is equal to $N \times N \times L$ where N is the spatial window size and L is the total frame length. In order to attain structure information in both spatial and time domain, this volume is subdivided into $n_x \times n_y$ cells in spatial domain and n_t cells in time domain.

From the experimental results, we have observed that because of the decreasing the quality of frame with Gaussian blur, interest point can vanish. Hence, the correct trajectory estimation may not be obtained for long trajectory length L. Empirically, several parameters are selected as $N = 32$, $L = 6$, $n_x = 2$, $n_y = 2$ and $n_t = 2$.

In the following parts, we will explain two feature extraction methods on trajectory including histogram of orientated gradient (HoG) [32] and motion boundary histogram [29, 33].

2.2.2.1 Histogram of Orientated Gradient (HoG)

Firstly, histogram of orientated gradient [32] is proposed for human detection. Later, this method is applied on action recognition with spatio-temporal feature extraction [36]. These studies validate that this representation gives superior performance results on action recognition problem.

The idea is, similar to scale invariant feature transform (SIFT), single 8 bin gradient orientation histogram is created by weighting with gradient magnitudes for each cell. Then, an L_2 normalization is employed over this histogram. The dimension of final feature vector is equal to $n_x \times n_y \times n_t \times 8$, in our setup $2 \times 2 \times 2 \times 8 = 64$.

2.2.2.2 Motion Boundary Histogram (MBH)

Authors state that, optical flow field between two consecutive frames composes of background, foreground and even camera motion. However, in motion feature, the camera motion including tilting, zooming etc. reduces the distinctive power of feature.

The underlying assumption [29, 33] is the typical camera motion contains local translations in other words, motion flows in neighborhood area behave coherently. Thus, the derivation on horizontal and vertical axis would discard the regular motion and yield the absolute motion for that point.

First, the proposed method splits the optical flow field into horizontal and vertical components. In 3×3 spatial window, the derivation are computed for each pixel and orientation histograms are constructed on each axis. The magnitude of motion is used for weighting these histogram. Both histograms are normalized separately. Since there are two separate histograms with dimension of $2 \times 2 \times 2 \times 8 = 64$, these two vectors are concatenated at the end and the final dimension of feature is equal 128.

CHAPTER 3

QUANTIZATION-BASED INDEXING

Although local descriptors induce an excellent representation for a frame thereby a video because of their invariance against occlusion clutter and geometric transformation, adequate number of descriptors need to be extracted from a video in order to make robust similarity search. This amount is generally huge and it obstructs to compare these descriptors within acceptable time range. Mapping feature into lower dimension would be an acceptable solution to reduce the comparison complexity and memory requirement.

During mapping feature vectors into lower dimension; there will be some information loss. Hence, for effective representation, the trade-off between comparison complexity and amount of information loss should be adjusted well.

Simplest way to map a descriptor vector into lower dimension is multiplying by an orthogonal projection matrix [48]. Since the projection is employed by multiplication of descriptor vector and projection matrix; the intended vector length can be easily adjusted by matrix dimensions. However, this assumption maps input vector directly to lower dimensional space without explicitly investigating any prior information about components of descriptor vector.

The assumption [37, 38] is that each component of feature vector cannot comprise similar amount of information. Hence, by the help of this assumption, dimension reduction can be improved even further. For that purpose, a feature corpus that consists of either supervised or unsupervised data is utilized to compute the contribution of each component. Leveraging different priority of each component, dimension reduction would be achieved by retaining top N most prior components with less information loss.

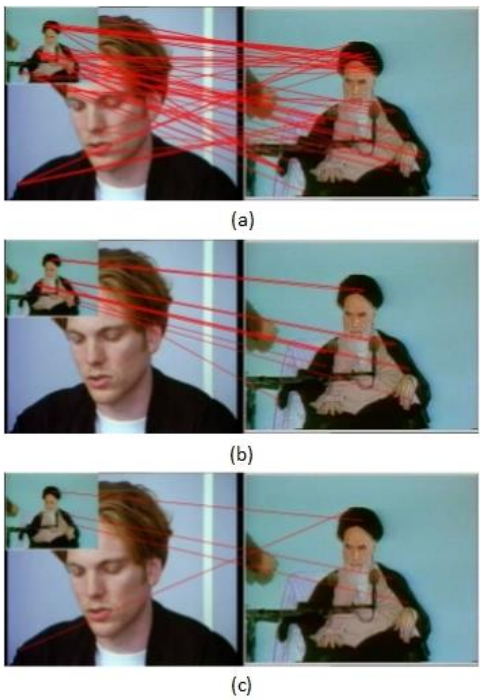

Figure 7. Matching query and reference points according to their indices. (a) Bag-of-Word, (b) Hamming Embedding, (c) Product Quantization

The main deficiency of these types of methods is that even if seamless dimension reduction is achieved, similarity search still takes place exhaustively. However, comparing these features whose contents are too different from each other leads unnecessary calculation. Another limitation is, in order to avoid curse-of-dimensionality, the feature whose dimension is higher than 10 should not be mapped with these methods.

In the recent years, due to the effectiveness of fast search capability and ease of implementation, quantization-based indexing procedures are frequently deployed for object recognition [43] and large scale image search [42, 51]. The underlying assumption is that feature vectors are represented with a set of indices by quantizing vectors in a pre-clustered space.

Within the scope of this thesis, we have investigated three state-of-the-art quantization-based indexing methods from simple to complicated one, namely bag-of-word [45], hamming

embedding [48] and product quantization [49]. Additionally, inverted index structure [45] and weighting signatures according to their term frequencies [45] have been explained in detail. Thus, we have divided this chapter into two main sections. In the first section, we will mention about these three indexing methods and we will give broad explanations about each method separately. Then, we will explain and discuss the effects of inverted index [45] and term frequency weighting [45] on detection accuracy and speed.

3.1 Content Indexing

With the availability of large video and image collections, making effective search while conserving effectiveness on large collections appeared as a challenging problem. According to the recent studies [45], quantization-based indexing schemes have come into prominence due to its scalability and robustness. Especially, the success of representing image/video with local patches has a foremost influence on this improvement [25-28].

Literature on this approach was primarily adapted from text-domain [34]. Because of success of this representation, the studies have been turned into visual domain and state-of-the-art methods have been proposed that particularly aim two necessities of representation; discriminative power and speed.

In the following sections, we will explain three existent and state-of-the-art quantization-based indexing methods. Bag-of-word scheme [45] represents feature vector with indices by making a quantization on a set of pre-clustered centers according to distance metric. Since clustering scheme is an unsupervised machine learning solution [52], selection of best-fitted cluster center size is a fundamental problem and this parameter designates the trade-off between robustness versus discriminative power.

In hamming embedding scheme [48], location of the feature vector inside the corresponding cluster is approximately encoded by a binary signature in addition to closest cluster center index. In product quantization [49], in order to augment bit code per feature component during

quantization, feature vector is uniformly partitioned into sub-vectors. Leveraging this assumption, residual error between the vector and corresponding cluster center is encoded.

Figure 7 illustrates the feature vector matching stage by their indices between query and reference frames for three quantization-based indexing schemes.

Although all these methods follow similar vector quantization scheme, each one has its unique diversity in similarity metric. To improve understandability, unique similarity metric formula for each method is denoted as $s_{model}(v^r, v^q)$ where v^r and v^q represent reference and query feature vectors.

3.1.1 Bag-of-Word

This phenomenon was firstly unveiled for text categorization and retrieval [34]. Mainly, it consists of two stages that are learning list of targeted stems and altering input vector to those stems according to similarity distances.

First, the procedure parses document into words and then these words are transformed into meaningful stems consecutively. Second, correspondences of stems are obtained from the learning list and document is represented with a vector where each bin is equal to accumulation of occurrence frequency of the words that are in learning list.

Insertion of term frequency and inverted index structure [45] that we will investigate in Section 3.2, make also tremendous improvement on detection accuracy and calculation speed.

Because of the superiority and simplicity of this method, it was explored in computer vision field firstly by Sivic and Zisserman [45]. In the proposed method, each of the stages in text domain associates with visual analogy. For example, instead of text stems, sparsely sampled scale, rotation and affine invariant patches are admitted and these patches are represented with the indices that indicates corresponding closest cluster center in pre-clustered vector space.

In the proposed method, initially, a coherent corpus is created from logically and randomly sampled d dimensional feature vectors $\boldsymbol{V} = \{v_1, v_2, \dots, v_N\} \in \mathbb{R}^{d \times N}$. Then this corpus is clustered into K non-overlapping regions $\boldsymbol{C} = \{c_1, c_2, \dots, c_K\} \in \mathbb{R}^{d \times K}$ and a fixed-sized clustered vector space is obtained for future vector alteration. In vision domain, vector space and each cluster center are named as *visual codebook* and *visual word* respectively [45].

Frequently, k-means algorithm [52, 53] is utilized for clustering stage. Briefly, this method consists of two steps namely *assignment* and *update* and it refines cluster centers iteratively [52, 53]. In assignment step, each visual sample in the corpus assigns to closest cluster centers. Even though different similarity metric have different effects on sorting of cluster centers, in general, euclidean distance metric (16) is used. In update step, means of each cluster are updated according to these samples. These two steps are repeated until the cluster centers converge implicitly or exceed a maximum number of iteration.

$$euclidean(v^1, v^2) = \sqrt{\sum_{i=1}^{d} ({v^1}_i - {v^2}_i)^2} \tag{16}$$

where ${v^1}_i$ and ${v^2}_i$ are the i the element of vectors.

After the construction of visual codebook that holds a set of visual words, ownership of the input feature vector is computed on these cluster centers. Theoretically, the underlying thought of this methodology is to represent feature vector v with a corresponding closest cluster center $q_c(v)$ in the visual codebook. In order to map the vector into indices, first, similarity distance between the vector and all visual words need to be calculated. Then, the closest visual word indicates the corresponding indices of the vector in this visual codebook.

Commonly, for scene understanding and object recognition tasks, bag-of-word is utilized by computing index occurrence histogram to obtain the unique overall structures of different

object and scene models [46, 47]. However, in content similarity search, our aim is to find same patches under different geometric transformation and noise models on reference frame. Hence, descriptor comparisons according to their indices equality is enough. Thus, the feature vector mapping stage can be introduced as a discrete optimization problem with several constraints as:

$$\begin{aligned} & arg\min_{\boldsymbol{B}} \sum_{i=1}^{N} \left\| v^i - \boldsymbol{C} b^i \right\|^2 \\ & st.\ \left\| b^i \right\|_{l^0} = 1, \left\| b^i \right\|_{l^1} = 1, b^i \geq 0, \forall i \end{aligned} \tag{17}$$

where $\boldsymbol{C}$ is the visual word list, v^i is the feature vector and b^i denotes the ownership of the feature vector on visual words as one or zero. The aim is to obtain b^i vector by optimizing the distance between the vector and cluster centers. As k-means algorithm assigns the feature vector implicitly into single index, these constraints emphasize that only one component of b^i should be equal to one, while remaining is zero.

Due to the fact that distance calculations are repeated for all cluster centers, for large value of K, huge computation time is spent on assignment of indices. Hence, iteration-based closest cluster center estimation becomes useless. This deficiency has triggered a demand on exploring an efficient search procedure. The recently adapted space search technique Kd-Tree [51] appears as a suitable solution for this problem. The underlying assumption is, instead of searching the vector within cluster centers one by one, partitioning cluster centers into sub-tree structures would narrow the search space.

To construct the tree-based structure, cluster centers are split recursively into two nodes leveraging two notions as inclusion of the dimension with the highest variance and the median value along the dimension. Hence, the redundant comparisons are discarded to provide fast search ability. Detail about this method can be found in [51].

Since this technique represents feature space with a set of tree-based structures, there might be error during assignment step particularly on the vectors that are close to cluster boundaries. Therefore, this type of search space techniques is categorized as approximate nearest neighbor method [51].

In bag-of-word method, due to the fact that each descriptor vector is represented with single quantizer $q_c(v)$, constituting a relation among query and reference feature is merely based on controlling equality of indices. Therefore similarity score formulation forms as:

$$s_{Bow}(v^r, v^q) = \begin{cases} 1.0, & if\ q_c(v^r) == q_c(v^q) \\ 0.0, & otherwise \end{cases} \tag{18}$$

Verbally, quantized indices must have same values to assert that these two descriptors are identical. However since coverage of each cluster center is huge particularly for small *K* in feature space, there may be a significant quantization error.

3.1.2 Hamming Embedding

The critical parameter of quantization-based approaches appears as *K* value. The reason is that *K* determines the trade-off between robustness and distinctivity. Roughly, for a small value of *K*, the probability of residing noisy version of descriptor in the same cell is high. However, this generates a weakness and the descriptors, which may contain irrelevant content, can be labeled with same indices. Conversely, when *K* value is high, homogeneity of cluster content is also high and precise estimation can be made. However this time, the probability of assigning noisy feature to same cluster is low.

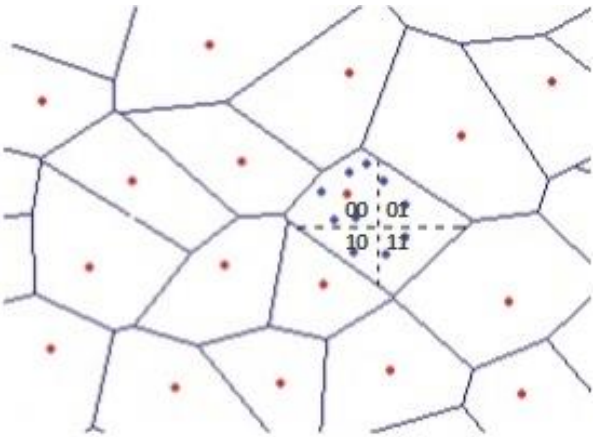

Figure 8. Red and blue dots indicates cluster center and sample points in two dimensional feature space respectively. Hamming embedding partitions each cluster region and encodes with a binary signature.

Based on this weakness, Jegou et al [48] propose a novel method that combines the advantages of coarse (lower K) and fine (higher K) quantizers. The assumption is that besides the quantized indices $q_c(v)$, location of feature vector within the cluster center is also encoded with a binary signature $b_{he}(v) = \{b_1(v), b_2(v), \dots, b_{d_b}(v)\}$ where d_b is the length of binary signature.

By the help of this binary signature, each cluster is quantized once more into sub-sectors as shown in Figure 8. Translations between these sectors are permitted with an error metric which corresponds to hamming distance as:

$$H_{he}(v^r, v^q) = \sum_{i=1}^{d_b} |b_i(v^r) - b_i(v^q)| \tag{19}$$

For binary signature estimation, this method can be reserved in two stages. First, in order to embed sub-sectors information into the visual codebook, necessarily parameters are obtained in offline learning stage. In the second stage, binary signature is computed exploiting these parameters for the given input vector.

Since a binary value needs to be computed for each component of feature vector, direct use of the vector causes unnecessary dimensional increase of signature. Hence, feature vector should map to lower dimension by multiplying orthogonal projection matrix $\boldsymbol{P}$ firstly. The dimension of matrix must be equal to dxd_b where d is length of descriptor vector and d_b is the dimension of intended lower space. In order to generate an orthogonal projection $\boldsymbol{P}$ matrix, QR factorization is applied on randomly sampled matrix of Gaussian values [48]. In this work, we set d_b as 32 for all feature vector models.

In the first stage of the method, to modify visual codebook and learn a set of parameters that enables us to create a binary signature, a feature vector corpus is build $\boldsymbol{V} = \{v_1, v_2, \dots, v_N\} \in \mathbb{R}^{d \times N}$ similar to codebook construction. Then, index of a vector is obtained conventionally assigning to closest visual word $q_c(v)$ and the vector is projected onto a set of components $\boldsymbol{Z} = \left\{z_{q_c(\boldsymbol{v}),1}, z_{q_c(\boldsymbol{v}),2}, \dots, z_{q_c(\boldsymbol{v}),d_b}\right\}$ by $\boldsymbol{P}$ matrix. These quantization and projection steps repeated for all feature vectors in the corpus. In order to use these parameters in further calculations, these projected components are stored alongside of cluster center indices.

Since location information of the vector inside cluster is encoded by a binary string or signature, each binary value has two options for representation. Therefore, an unique threshold $\tau_{c,h}$ is computed for each projected component of the cluster where $c = 1 \dots K$ and $h = 1 \dots d_b$. This threshold value corresponds to median value among all the estimated components for cluster c in feature corpus. For future signature calculation, projection matrix $\boldsymbol{P}$, median values for each cluster and component $\tau_{c,h}$, are stored in addition to visual words.

In the second stage, to estimate a binary signature, first, descriptor vector is assigned to the closest visual word $q_c(\boldsymbol{v})$. Then this vector is projected to lower dimensional space using $\boldsymbol{P}$ matrix and a set of components $\boldsymbol{Z} = \left\{z_{q_c(\boldsymbol{v}),1}, z_{q_c(\boldsymbol{v}),2}, \dots, z_{q_c(\boldsymbol{v}),d_b}\right\}$ is calculated. To obtain the binary signature $b_{he}(v) = \left\{b_1(v), b_2(v), \dots, b_{d_b}(v)\right\}$, each projected component is compared with $\tau_{q_c(v),h}$ separately as:

$$b_i(v) = \begin{cases} 1, & if\ z_{q_c(v),i} > \tau_{q_c(v),i} \\ 0, & otherwise \end{cases} \tag{20}$$

Besides the quantization index value of the vector, in order to measure the similarity between two feature vectors, approximate localization information with binary signatures would be utilized to refine the accuracy. As defined in [48], hardcoded similarity score function can be given as:

$$s_{HE}(v^r, v^q) = \begin{cases} 1.0, & if\ \begin{matrix} q_c(v^r) == q_c(v^q) \\ H_{he}(v^r, v^q) < h_t \end{matrix} \\ 0.0, & otherwise \end{cases} \tag{21}$$

where h_t denotes the threshold which can vary in range of $0 \leq h_t \leq d_b$.

However, in our case, descriptor may contain noise because of re-encoding attacks. Empirically, we observed that instead of selecting hardcoded similarity score as in (21), weighing similarity score according to hamming distance of the binary signatures would yield better performance. Therefore, we have reintroduced similarity score formula according to this observation as:

$$s_{HE}(v^r, v^q) = \begin{cases} 1.0 - \dfrac{H_{he}(v^r, v^q)}{h_t}, & if\ \begin{matrix} q_c(v^r) == q_c(v^q) \\ H_{he}(v^r, v^q) < h_t \end{matrix} \\ 0.0, & otherwise \end{cases} \tag{22}$$

The hardcoded threshold h_t is selected as 22 in this work. Briefly, the influence of weighting scheme with hamming distance is to favor small distanced pairs with higher similarity score.

3.1.3 Product Quantization

The purpose of quantization scheme is to represent feature vector with a few parameters while preserving high discriminative power as much as possible. Hence, reserving high bit rate for

each component of vector is directly related with high cluster size K. Inherently, this affects positively the success rate of retrieving. However, the increase in number of required sample and complexity of learning induces a limitation to obtain an effective quantizer. Additionally, as K and d denote the visual word size and the dimension of descriptor vector respectively, $K \times d$ floating point value need to be stored for each quantizer which makes the approach impractical for large cluster size.

Product quantization scheme can be presented as a solution to reach high cluster size without implicitly following clustering procedure. Initially, this approach has been extensively studied in information theory and it have been tailored by [49] into machine learning field. The underlying assumption is that instead of employing a vector v as whole, vector is split into m uniform sub-vectors v_m where $1 \leq m \leq M$ and length of each sub-vector is $d^* = d/m$. The quantization step is done for each sub-vector separately using m different quantizers $q_m(.)$ as in:

$$\{v_1, v_2, .., v_m\} \rightarrow \{q_1(v_1), q_2(v_2), \dots, q_m(v_m)\} \tag{23}$$

Since the aim of product quantizer is to enlarge cluster size K by combining several sub-vector quantizers $q_m(v_m)$, the final cluster size K explicitly is equal to $(K^*)^m$ where K^* is the cluster size for each sub-quantizer $q_m(.)$.

Additionally, the complexity of learning and assignment of product quantizer are decreased to $m \times K^{1/m} \times (d/m) = K^{1/m} \times d$ whereas original K-means algorithm complexity is equal to $K \times d$.

To conduct this method on large scale visual search, similar to hamming embedding, a small code for each feature vector is inserted in addition to the quantized vector index. This code encodes the residual error between the vector and corresponding cluster center (24) with product quantizer. Thus, this technique improves the accuracy more precisely than utilizing purely vector index itself.

$$r(v) = v - q_c(v) \tag{24}$$

This method also needs a pre-configuration step to obtain necessary parameters. First, a corpus of residual error is constructed from randomly sampled image feature vectors. Then, owing to the underlying assumption of product quantization, each residual vector is split into m uniform sub-vectors and these sub-vectors are clustered with k-means algorithm separately into K^* cluster centers. This clustering step is repeated for all m partitions. To facilitate the future calculations, a look-up table which stores the distance and order relations of cluster centers with each other is generated for each partition.

In assignment stage, first, in order to obtain the small code, residual error $r(v)$ is computed with (24) and it is quantized with product quantizer $q_p(r(v))$ into m distinct partitions $q_m(r(v_m))$ where $m = 1 \dots M$. Thus, by the help of this method, the vector is represented approximately as:

$$\tilde{v} = q_c(v) - q_p\big(v - q_c(v)\big) \tag{25}$$

In this method, to calculate the soft-assignment of similarity score of two signatures, we apply two constraints that should be ensured implicitly, otherwise it will be accepted as zero. First, the quantized indices of two vector should be identical similar to bag-of-word method. Secondly, unlike [49], we utilize a soft similarity score. Our metric uses the order of the quantization query sub-residue $q_m\big(r({v^q}_m)\big)$ in nearest neighbors of the quantized reference $q_m\big(r({v^r}_m)\big)$. We take into account nearest neighbors up to τ_{pq} neighbor which is empirically determined. By using this order relation, we define the similarity score of the method as:

$$s_{HE}(v^r, v^q) = \frac{1}{M} \sum_{1 \le m \le M} 1.0 - \frac{NN_m\left(q_m\big(r({v^r}_m)\big) \middle| q_m\big(r({v^q}_m)\big)\right)}{\tau_{pq}} \tag{26}$$

where $NN_m(.)$ gives the order of $q_m\left(r({v^q}_m)\right)$ in nearest neighbors of $q_m\left(r({v^r}_m)\right)$ for m^{th} sub-residue vector. In this work, K^* and τ_{pq} are selected as 256 and 50 respectively. In order to provide similar condition for all feature extraction methods whose vector dimension is different, m is determined according to feature vector size.

3.2 Inverted Index Structure and Term Frequency Weighting

The similarity search of the feature vectors on two images should take place exhaustively. However, comparing the feature vectors whose quantization indices are different generates a redundancy. For example, as K and L denotes the total number of local descriptors on reference and query frames, at least $K \times L$ comparison should be made in order to find true correspondences. However, for large dataset, this becomes nearly impossible.

By the influence of quantization-based indexing methods, each feature vector is represented with an index. Therefore, comparing only the descriptors whose indices are same would facilitate the efficiency of retrieval.

This thought is very common in database systems and text-based search engines in order to accelerate the response to user. Thus, each element of reference archive is stored in a data structure entry according to its index value that can be either words or numbers.

Similar structure can be utilized on visual domain leveraging quantization-based indexing scheme [45]. The underlying assumption is that each descriptor can be stored in a data structure according to its visual word id that are obtained by quantization-based methods.

Hence, in this structure, the total entry number is equal to visual word size. Also, due to the fact that increment of K extends the possibility of assigning descriptors on different entries, the comparison speed would be accelerated.

To improve the accuracy of detection, weighting indices has positive influence on accuracy. In text domain [34], rare words would contain more distinctive information rather than frequent ones, the underlying assumption is that stop words which are frequently occurred in text like ‘the’ or ‘a/an’, should be discarded or weighted according to their importance.

This weighting scheme combines two statistical intuitions including term frequency and inverted index frequency [34, 45]. Term frequency and inverted index frequency weight the word according to its occurrence in a particular image and in a database respectively as:

$$w_{tfidf} = \frac{n_{c,d}}{n_d} \times log \frac{N}{n_c} \tag{27}$$

where $n_{c,d}$ is the number of occurrences of word c in document d, n_d is the total number of words in document d, n_c is the number of occurrence of word c in whole database and N is the total number of document in dataset.

CHAPTER 4

GEOMETRIC VERIFICATION

The main deficiency of deploying quantization-based indexing methods on local descriptor is, in ranking stage, geometric consistency that exists among local patches is discarded. Hence, that causes an ambiguity in matching stage and decreases the accuracy of detection drastically. This information is frequently reintroduced by adding a re-ranking stage that computes the geometric transformation between matched correspondences in reference and query points. Additionally, encoding local relation of spatial patches has a positive influence on mitigating this ambiguity.

Generally, in order to constitute a geometric consistency on entire frame, 2D homography [50] or its iterative versions [51] are utilized. The assumption is that Hough estimates a transformation with four degree of freedom and each pair of matches generates a set of parameters. Later the set of matches from largest bins are used to estimate a finer 2D transformation.

Despite faster hardware and code optimization, homography estimation cannot apply on more than a small set of top results in the initial ranking stage due to its computation cost.

Local geometric relation [56] is exploited with the spatial relation of interest points by frequently leveraging co-occurrences of visual words in neighborhood area. However, complex comparison scheme on neighboring words induces a deceleration on calculation speed with boosting memory usage.

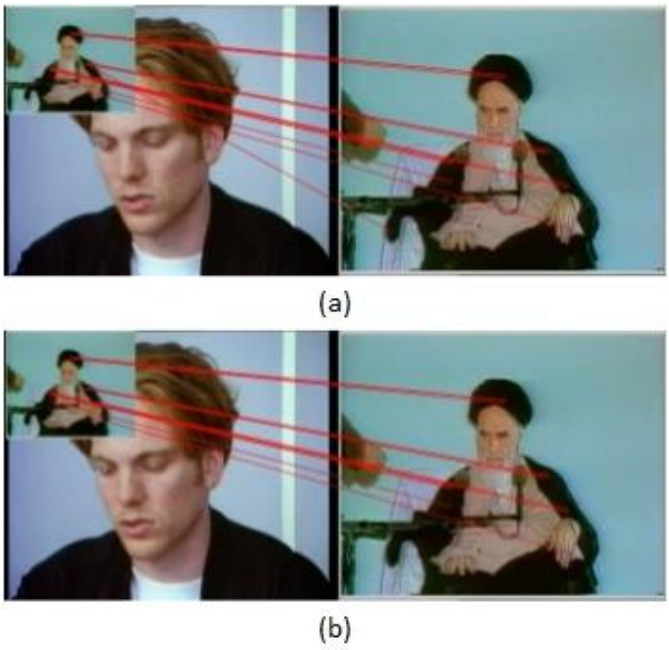

Figure 9. Elimination of outliers by constituting a geometric consistency. (a) Bag-of-Word, (b) Bag-of-Word with Weak Geometric Consistency.

Due to the requirement of high computational power and memory, these two approaches are not appropriate particularly considering immense data size on this task. Hence, convenient geometric verification stage should be as simple as possible to be applicable to the entire archive.

For all these reasons, we have devoted this chapter to investigate geometric verification techniques. Hence, this chapter is partitioned into two sections to discuss weak geometric consistency among corresponding pairs on the frames and fast and compact local geometric relation signature thoroughly. First, we will explain how to construct the approximate geometric transformation between the matched points for both spatial and spatio-temporal feature models in detail. Also we will derive all necessary mathematical formulations to improve the intelligibility.

Second, a novel geometric relation signature that is invariant to scale and orientation changes will be proposed. The underlying thought of this method is to encode the spatial relation of

interest points by merely checking existence or non-existence of the visual word in neighborhood area. Hence, this geometric relation enables us to compute a geometric signature which consists of a set of binary values. Thereby, it allows rapid bit comparisons besides providing essential information about local geometry of interest points.

4.1 Weak Geometric Consistency

In order to constitute a geometric consistency among matched correspondences in reference and query points, several spatial and/or temporal geometric characteristics can be utilized. These parameters can be scale, orientation and spatial coordinates of interest points. This refinement stage is frequently reintroduced at the end of the process as a filtering stage to eliminate mismatches (outliers) that are obtained from initial content signature matching as shown in Figure 9.

In literature, although state-of-the-art 2D homography approach is mostly preferred method to estimate true geometric transformation, it consumes huge computation power particularly on obtaining perfect geometry according to larger bin characteristic. Thus, this high computational cost limits the suitability of this method for large dataset.

Based on the limitation of homography estimation, proposed geometric consistency stage should be simple to carry out to entire archive while conserving high effectiveness and robustness.

In [48], authors investigate an approximate geometric consistency among matched points without explicitly verifying exact geometric transformation. Hence, these methods sacrifice the quality of estimation to extend the scope of influence.

This method [48] which pioneers this type of manner, obtains approximate geometric characteristic by purely exploiting orientation and scale changes. Simple assumption is that when an image undergoes rotation and scale changes, due to the fact that local descriptors are

also invariant, all the interest points on an image are substantially affected same amount. Hence, geometric consistency can be constituted by seeking a global characteristic distribution on scale and orientation changes.

Theoretically, to minimize the space of interest for both orientation θ^r and scale s^r parameters, first these parameters are quantized into q_{θ^r} and q_{s^r} which denote the uniform quantizer in orientation (28) and the logarithmic quantizer in scale (29) respectively.

$$q_{\theta^r} = \theta^r / qs_{step} \tag{28}$$

$$q_{s^r} = \log_2 s^r \tag{29}$$

where qs_{step} is the uniform quantization parameter. In this work, we set it as 8.

These procedures are repeated on query points q_{θ^q} and q_{s^q} as well. Then, in order to obtain the approximate geometric transformation for all correspondences on query and reference frames, distribution of orientation (30) and scale (31) differences are computed. At the end, two individual distribution histograms referring to $h^{\tilde{s}}$ and $h^{\tilde{\theta}}$ are constructed for scale and orientation.

$$\tilde{\theta} = q_{\theta^r} - q_{\theta^q} \tag{30}$$

$$\tilde{s} = q_{s^r} - q_{s^q} \tag{31}$$

The purpose of utilizing two separate distribution histogram is to minimize the cost of memory allocation. According to author's statements, these two histograms appear as marginal probabilities of 2D histogram and this assumption is as precise as full 2D histogram. In this method, the final similarity score of two frames is equal to minimum value of two histogram's maximums as:

$$s_{wgc} = min\left(max\left(h^{\tilde{s}}\right), max\left(h^{\tilde{\theta}}\right)\right) \tag{32}$$

Under this motivation, we have reserved this section into two parts to investigate geometric consistency for spatial and spatio-temporal signatures separately. In these parts, instead of scale and orientation changes, we will exploit translation characteristic of points. For spatio-temporal signatures, a novel method is proposed by adding temporal behaviors of trajectories in addition to spatial properties.

4.1.1 Spatial Geometric Consistency

Even though true local descriptor matching is achieved seamlessly, there would be outliers that reduce the accuracy of detection. Hence, utilization of geometric consistency by computing approximate transformation would eliminate incorrect matches.

In spatial domain, the formulation of spatial transformation among two matched points r and q (reference and query points respectively) depends on scale factor s, orientation parameter θ, 2D spatial positions of (x^r, y^r) , (x^q, y^q) and translation characteristic $[t_x, t_y]^t$ as:

$$\begin{bmatrix} x^q \\ y^q \end{bmatrix} = s \times \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} \times \begin{bmatrix} x^r \\ y^r \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \tag{33}$$

Although the proposed method in [54] constitutes simple but convenient geometric consistency by merely exploiting scale and orientation parameters, this type of approaches still has a space for further improvements due to the fact that scale and orientation characteristics are not discriminative enough.

In [54], authors propose an enhancement for approximate spatial consistency including translation information. The idea is, since translation characteristic still considers the scale and the orientation characteristics as in (33), inclusion of translation distribution provides extra geometric clue.

First, similar to previous method, scale s^r and orientation θ^r parameters of reference descriptor are quantized into q_{θ^r} (28) and q_{s^r} (29) (q_{θ^q} and q_{s^q} for query descriptor.). Differently, (34) is employed to model the scale changes due to the logarithmic operation.

$$\tilde{s} = 2^{(q_{s^r} - q_{s^q})} \tag{34}$$

Then, scale and orientation changes in (34) and (30) equations are combined with (33) and spatial transformation formulation is reintroduced as:

$$\begin{bmatrix} \tilde{x}^q \\ \tilde{y}^q \end{bmatrix} = \tilde{s} \times \begin{bmatrix} cos\tilde{\theta} & -sin\tilde{\theta} \\ sin\tilde{\theta} & cos\tilde{\theta} \end{bmatrix} \times \begin{bmatrix} x^r \\ y^r \end{bmatrix} \tag{35}$$

where $\tilde{x}^q$ and $\tilde{y}^q$ are approximate corresponding locations of spatial transformations of x^r and y^r coordinates according to characteristic changes in scale $\tilde{s}$ and orientation $\tilde{\theta}$. For query point q, the translation difference $\tilde{t}$ is calculated as:

$$\tilde{t} = \sqrt{(\tilde{x}^q - x^q)^2 + (\tilde{y}^q - y^q)^2} \tag{36}$$

Then, the translation differences for all matched descriptors are computed and a translation histogram h^t is constructed. This histogram holds the distribution of scores according to corresponding translation characteristic of matched pairs. At the end, an aggregation is expected on a single bin of the distribution histogram for duplicate images. To enhance the robustness, authors propose a smoothing operation by averaging two neighborhood bins of peak p as;

$$\tau_{peak} = \left|h_p^t\right| + \left|h_{p+1}^t\right| + \left|h_{p-1}^t\right| - 2 \times \frac{\sum_{i=1}^{m} \left|h_i^t\right|}{m} \tag{37}$$

where m is the number of histogram bins and $p + 1$ and $p - 1$ are the neighboring points.

Since rotation of frame is not commonly encountered in copy detection, in order to accelerate the calculation speed, we replace investigating distribution of orientation characteristic with a simpler constraint. According to this constraint, uniformly quantized orientations for both query q_{θ^q} and reference q_{θ^r} descriptors should be identical to continue for further voting scheme. Although this constraint causes a quantization error, it accelerates comparison stage by immediately discarding the points that do not satisfy.

With this constraint, orientation difference in (33) will be equal to zero. Hence, orientation matrix turns into identity matrix. The formula simplify as:

$$\begin{bmatrix} \tilde{x}^q \\ \tilde{y}^q \end{bmatrix} = \tilde{s} \times \begin{bmatrix} x^r \\ y^r \end{bmatrix} \tag{38}$$

Empirically, in order to improve accuracy and efficiency of geometric consistency, we have replaced couple of procedures that are proposed in [54]. Differently, we have reintroduced the calculation of translation difference for point q with Manhattan distance (39) instead of L_2 norm to accelerate the calculation speed.

$$\tilde{t} = |x^q - \tilde{x}^q| + |y^q - \tilde{y}^q| \tag{39}$$

Additionally, we have opted to establish a 2D distribution histogram $h^{\tilde{t}}$ instead of two separate 1D histograms which enables us to investigate scale and translation changes at the same time. That is we treat these two distributions as a joint probability density.

Thirdly, although joint usage of scale and orientation changes in 2D histogram yields better results, it boosts memory allocation and impedes the calculation that is spent on estimating the peak value of the histogram. Hence, we have mitigated this deficiency by utilizing an uniform quantizer for translation difference q_t which helps to reduce the space of interest as:

$$q_t = \tilde{t}/qt_{step} \tag{40}$$

where qt_{step} is the quantization parameter. In this work, it is selected as 20. We have observed that averaging peak bin with neighboring ones induces a decrease on accuracy in our assumption due to utilizing a quantizer in advance. Hence, final similarity score is equal to the maximum value of 2D histogram as:

$$s_{twgc} = \max\left(h^{\tilde{t}}\right) \tag{41}$$

Weak geometric consistency with scale and orientation differences is invariant to flip transformation. However, exploiting translation difference is not invariant to this type of transformation and it can induce an increase miss rate. Hence, proposed method should be aware of this kind of attack to increase the success rate.

In our case, due to the hardness of distinguishing flipped image from original one, frequently mirror attacks have been introduced on vertical axis. Thus, this transformation ruins the geometry of points that is proposed on (39) and the consistency becomes useless.

For that purpose, geometric consistency should be customized for this problem completely. Flipping image vertically, x coordinate of the interest point is deformed as $width - x$ where $width$ is the width of the frame. If we place this coordinate of interest points in (38), it will be equal to:

$$\begin{bmatrix} width - \tilde{x}^q \\ \tilde{y}^q \end{bmatrix} = \tilde{s} \times \begin{bmatrix} x^r \\ y^r \end{bmatrix} \tag{42}$$

Similarly, the translation difference for point q is formed as:

$$t = |x^q + \tilde{s} \times x^r - width| + |y^q - \tilde{s} \times y^r| \tag{43}$$

Since $width$ is constant, it can be discarded from the formulation (43). Final translation difference can be found as:

$$t = |x^q + \tilde{s} \times x^r| + |y^q - \tilde{s} \times y^r| \tag{44}$$

Since there is no prior information about query frame is whether flipped or not, proposed method should handle both cases successfully. Therefore, due to the disjoint relation, we have utilized two different 2D distribution histograms for original ${h_+}^{\tilde{t}}$ and flip version ${h_-}^{\tilde{t}}$ even it causes an increase on complexity.

The idea is that if two images are similar whether one of them is flipped or not, one of them would have a salient bin which yields the geometric characteristic of two images. Hence, even though two histograms are combined together, while scores in one histogram are scattered onto different bins, scores in other histogram aggregate on single bin for duplicate frames is expected. Therefore we have formed final geometric consistency score as:

$$s_{twgc} = \max\left({h_+}^{\tilde{t}}, {h_-}^{\tilde{t}}\right) \tag{45}$$

4.1.2 Novel Trajectory-Based Geometric Consistency

Since spatio-temporal signatures are computed on consecutive frames, they have unique geometric characteristics different from spatial domain. The spatial variations of interest points in time can be employed as a geometric clue to constitute a consistency among correspondence.

Spatio-temporal signature k in time sequence t_i has a set of spatial information including scale s and spatial positions $({x^k}_{t_i}, {y^k}_{t_i})$ for tracked points. Even if spatial information of these points can be utilized for each frame separately, it boosts the computation complexity.

Thus, these coordinates and variations in time should be modeled as a single compact feature, in order to establish a geometric consistency more effectively.

Hence, each trajectory is represented with four parameters as spatial means μ_x, μ_y and variances σ_x, σ_y of x and y where $\mu_x = \frac{1}{L}\sum_{i=1}^{L} x_i$ and $\sigma_x = \frac{1}{L}\sum_{i=1}^{L}(x_i - \mu_x)^2$ in addition to scale parameter which is constant during signature extraction. In here, L is the length of trajectory in time and similarly, μ_y and σ_y are computed for y axis.

By the help of these geometric parameters, we propose a novel geometric consistency method for spatio-temporal domain that consists of two steps. In the first step, spatial variances are considered. The underlying assumption is that if two corresponding pairs are identical, then their spatial variances should be roughly proportional with scale as given below (46).

$$\left|\sigma_{x,y}^{q} - \tilde{s} \times \sigma_{x,y}^{q}\right| < \tau_{\sigma} \tag{46}$$

where

$$\tilde{s} = \sqrt{2} \times (s^{q} - s^{r}) \tag{47}$$

Similar to spatial geometric consistency method, to reduce complexity, Manhattan distance is used to estimate the scatter between spatial variances of query and reference points. Therefore, $\sigma_{x,y}^{q}$ and $\sigma_{x,y}^{q}$ denote the summation of spatial variances on x and y axis for query and reference points as in equations (48) and (49) respectively. Additionally, s^{q} and s^{r} show the scale levels of query and reference points and $\tilde{s}$ indicates the scale level difference that is weighted with downsampling step parameter (In this work, it is selected as $\sqrt{2}$). Since some error might be occurred during the quantization of parameters, variance difference should be within a margin of misalignment τ_{σ} which is set as 3.

$$\sigma_{x,y}^{q} = \sigma_{x}^{q} + \sigma_{y}^{q} \tag{48}$$

$$\sigma_{x,y}^{r} = \sigma_{x}^{r} + \sigma_{y}^{r} \tag{49}$$

In the second stage, similar to spatial domain, the geometric transformation can be written as (50) by replacing single frame coordinate points with spatial means of trajectory in (33).

$$\begin{bmatrix} \mu_x^q \\ \mu_y^q \end{bmatrix} = \tilde{s} \times \begin{bmatrix} \mu_x^r \\ \mu_y^r \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \tag{50}$$

where μ_x^q and μ_y^q are the spatial means of trajectory on x and y axis for query point (μ_x^r, μ_y^r resp. for reference point), t_x and t_y are the translation on x and y axis respectively. Since orientations of spatio-temporal signatures are accepted as zero, orientation matrix will be equal to identity matrix. Thus, translation difference of query and reference descriptors is obtained as;

$$t_\mu = \left| \mu_{x,y}^q - \tilde{s} \times \mu_{x,y}^r \right| \tag{51}$$

Similarly, Manhattan distance is utilized and $\mu_{x,y}^q$ and $\mu_{x,y}^r$ are the summation of the spatial means of trajectory on x and y axis for query point and reference points in equations (52) and (53) respectively.

$$\mu_{x,y}^{q} = \mu_{x}^{q} + \mu_{y}^{q} \tag{52}$$

$$\mu_{x,y}^{r} = \mu_{x}^{r} + \mu_{y}^{r} \tag{53}$$

In order to treat scale and translation changes as a joint probability, similar to the previous spatial domain, 2D histogram $h^{\tilde{t}}$ is constructed. Translation characteristic is also quantized into an index by the uniform quantizer , whose step size is set as 20 to reduce the search space. For final similarity score maximum of 2D histogram can be accepted as:

$$s_{twgc} = \max(h^{\tilde{t}}) \tag{54}$$

4.2 Novel Local Geometric Relation Signature

The geometric relation of interest points can be encoded with a signature like exploiting content of video. Hence, besides leveraging geometric consistency on entire frame, it can provide extra geometric clue about local behavior of interest points.

However, effective representation should assure two crucial requirements namely robustness and simplicity for large dataset.

We propose geometric signature that encodes geometric relation with a binary signature by merely checking existence or non-existence of interest points in neighborhood area of the central point. Due the fact that this area is contructed by leveraging scale and orientation property of central point, inherently, this representation is invariant several geometric transformations.

In literature, there are several methods that aim to extract geometric signatures exploiting interest point-based relation. Although these methods yields fair results on several visual image search datasets, they cannot supply these two requirements completely.

In [55], relations on local regions are imposed by graph-based representation for visible range satellite image categorization task. Due to sensitivity of interest points to illumination changes and lack of invariance against scale clutter, different connections might be obtained when image undergoes these kinds of geometric transformations. In another work [56], image is firstly partitioned into regular grid in spatial domain and geometric signatures are extracted combining visual word id with distribution of descriptors on these grids. Similar to previous method, these signatures lack scale invariance.

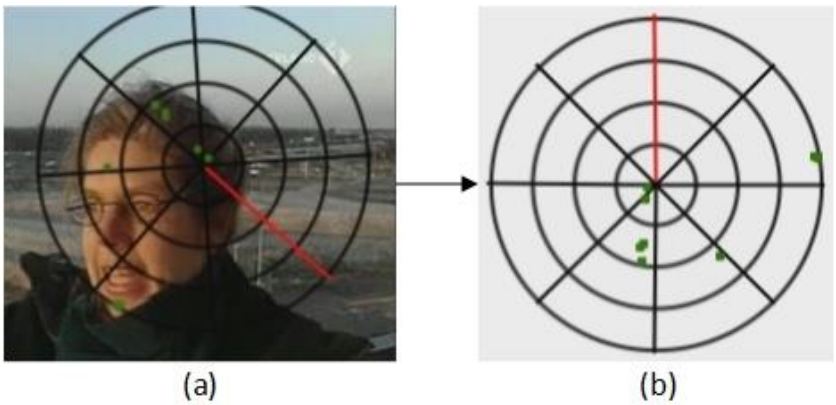

Figure 10. Computation of visual group binary signature. (a) Red line indicates the dominant angle and green dots denote the neighboring interest points. (b) The region is transformed according to its dominant angle.

In the source of the inspiration of our method [57], first, a circular region is defined around a central point and a geometric signature is encoded incorporating coordinate of neighboring visual words with their indices. In this case, the similarity of two signatures is directly proportional with true matching of neighboring visual words on coherent patches. However, for greater number of interest points in neighborhood area, computation complexity is multiplied alongside of memory. Thus, these representations are not applicable on this task.

The contribution of our proposed geometric signature is that instead of applying a complex voting scheme as in [57], the similarity score of two signatures is easily calculated with bitwise comparisons with a small burden of complexity.

In the proposed method, first, similar to [57], a circular region is defined around a master point that is explicitly the central interest point k exploiting its scale and orientation geometric characteristics. That makes this representation invariant to scale and orientation changes under the assumption that interest point detector is not affected from these transformations. Thus, for duplicate images, each one would contain identical geometric signatures.

Then, this circular region is partitioned into a set of patches G^k. To extract a visual group binary signature b_{vg}^k, from each patch $G_{i,j}^k$ where $i = 1 \dots N_{\delta_\theta}$ and $j = 1 \dots N_{\delta_s}$ are the number

of partitions in angular and scale domains respectively, a binary value is computed according to whether it contains any interest point or not. Then these binary values are concatenated in clock-wise manner from center to outer patches (55) as shown in Figure 10. The final length of the signature is equal to $N_{\delta_\theta} \times N_{\delta_s}$. In this work, empirically, we set N_{δ_θ} and N_{δ_s} as 8 and 4 respectively.

$$b_{vg}^{k} = \left\{ b_p\left(G_{1,1}^{k}\right), \dots, b_p\left(G_{N_{\delta_\theta}, N_{\delta_s}}^{k}\right) \right\} \tag{55}$$

where $b_p(.)$ is the visual group binary function as:

$$b_p\left(G_{i,j}^{k}\right) = \begin{cases} 1 & if\ any\ interest\ point\ exist\ in\ G_{i,j}^{k} \\ 0 & otherwise \end{cases} \tag{56}$$

The similarity score of two visual group binary signatures k and l (57) depends on summation of *AND* operation of same patches and a normalization parameter N_{norm}.

$$s_{vg}\left(b_{vg}^{k}, b_{vg}^{l}\right) = \frac{1}{N_{norm}} \sum_{\substack{1 \le i \le N_{\delta_\theta} \\ 1 \le j \le N_{\delta_s}}} b_p\left(G_{i,j}^{k}\right) \times b_p\left(G_{i,j}^{l}\right) \tag{57}$$

where N_{norm} whose value is equal to maximum of number of filled patches in G^k and G^l as:

$$N_{norm} = \max\left(arg\left(\sum_{b_{vg}==1} G^{k} \right), arg\left(\sum_{b_{vg}==1} G^{l} \right) \right) \tag{58}$$

In order to solve the lack of invariance of visual group binary signature against vertical flipping, for flip-detected descriptor, binary values are concatenated in counter clock-wise manner.

As expected, by adding this compact signature, local scale and orientation invariant relation is easily constituted and this scheme eliminates the outliers that might be obtained in content matching. However, for more accurate results, geometric consistency that explore the overall geometry of matches should be deployed besides this signature.

CHAPTER 5

EXPERIMENTAL WORK

In this chapter, we will analyze the performances of previously explained methods. The positive and negative aspects of each method will be investigated around three essential criteria for succeeding copy detection including high success rate, low computational complexity and low memory usage. For that purpose, TRECVID 2009 content-based video copy detection dataset [58] is selected for performance evaluation.

In the following sections, first, the details of dataset and evaluation metrics will be explained. Then, ranking stage for complete system and the performance scores for all methods will be presented using three evaluation metrics.

5.1 Dataset and Performance Metrics

We have utilized TRECVID 2009 content-based video copy detection dataset [58] in our experiments. The dataset consists of 400 hours reference videos and 1407 query videos. The length of each query video varies from 3 seconds to 3 minutes. 937 of query video are copied from this reference archive. In the dataset [58], in order to generate a query video, particular destructive operations are performed over some part of reference video as given below;

- Picture-in-picture (T2): Places spatially scaled reference video over another unrelated video.
- Insertion of pattern (T3): Inserts a pattern, text or banner.

Figure 11. Sample video frames from TRECVID 2009 dataset. (a) picture-in-picture (T2), (b) re-encoding (T4), (c) contrast changes (T6), (d) cropping, insertion of pattern and text (T8)

- Strong re-encoding (T4): Applies strong compression.
- Change of gamma (T5): Changes gamma parameter with different configurations.
- Decrease in quality (T6): Introduces combinations of blurring, frame dropping, gamma, contrast and white noise.
- Post Processing (T8): Introduces combinations of cropping, shifting, contrast changing, vertical flipping, insertion of pattern and picture-in-picture.
- Combination of 5 attacks (T10): Has randomly selected transformation from T2-T8.

Typical examples of several attack models are shown in Figure 11. Remaining of query videos is transformed from different video databases.

In pattern recognition and information retrieval, there are famous performance metrics that measure the accuracy of test in different aspects. Hence, in evaluation section, the performance scores will be presented in *recall*, *precision* and *f1-score* metrics.

Since recall measures the fraction of relevant documents that are successfully retrieved (59), the accuracy of each method will be computed using this metric for query videos that are copied from the reference archive.

$$recall = \frac{|\{relevant\ document\}| \cap |\{retrieved\ document\}|}{|\{relevant\ document\}|} \tag{59}$$

Additionally, the false alarm performance will be investigated with precision metric on the remaining query video. Precision metric corresponds to the fraction of retrieved document that are relevant (60).

$$precision = \frac{|\{relevant\ document\}| \cap |\{retrieved\ document\}|}{|\{retrieved\ document\}|} \tag{60}$$

In literature, the evaluation results are generally presented in NDCR [58] and f1-score metric for content-based video copy detection. Therefore, in order to compare the performance score of our developed system with competitors, we will give f1-score in addition to recall and precision scores. Briefly, f1-score considers the precision and recall scores jointly as given below:

$$f1score = 2 \times \frac{precision \times recall}{precision + recall} \tag{61}$$

5.2 Evaluation

In the previous chapters, we have summarized and proposed several algorithms for feature extraction, indexing and geometric verification stages. In this section, we will investigate

three essential criteria (that we gave at the beginning of this chapter) for each combination of these techniques, in order to obtain the most feasible system. Hence, first, we will propose overall structures of the developed system for both spatial and spatio-temporal signatures. Then, the details of each combination of feature extraction, quantization-based indexing, and geometric verification techniques will be described. At the end, the experimental results and observation will be presented according to results.

5.2.1 Ranking

In ranking stage, query video is searched in reference archive leveraging their coherent content characteristics. Thus, according to the similarity scores, an observation can be made whether the query video is copied from reference archive or not.

In the developed system, inverted index structure has been appended on each combination by default, because of the effectiveness and fast comparison capability. Empirically, we have observed that weighting content signature according to their term-frequencies would make significant improvements on detection accuracy. Thus, again by default, term-frequency weighting have been also inserted.

In order to index the large reference video archive in inverted indexing structure effectively, for each entry per descriptor following parameters are stored as given below:

- the video identifier $video_{id}$,
- the frame identifier $frame_{id}$,
- the additional indexing code (b_{he} if indexing method is hamming embedding or b_{pq} if indexing method is product quantization otherwise it is empty)
- the tf-idf weight of reference descriptor w_{tfidf},

To handle sufficient numbers of descriptors, all descriptors have been computed on one second interval frames to provide similar conditions for each local descriptor model. Therefore the similarity score between query and reference video is computed one-by-one frame comparisons. The aim is that if these two videos are duplicate, then there would be an accumulation on a bin of score vector that indicates the temporal alignment of query and reference video in time. Therefore following procedure is introduced to estimate the similarity score;

- Initiate a score vector whose size is equal to reference video duration. (Since we have sampled video in one second intervals)
- For each query frame, compute similarity score sc_{t_i,t_j} on each reference frame which corresponds to frame correlation in time domain and add this score to $t_i - t_j$ bin of score vector where t_i is the time instance of reference frame and t_j is the time instance of query frame.
- Maximum value of score vector is equal to the final similarity score and this bin yields temporal location of the match.

For single query video, these steps are repeated for all reference video and these scores are sorted in descending order. At the end, the detection rule is applied to first and second results which emphasizes that the score of the first result should be at least twice of the second result in order to accept the first result as a copy.

Spatial and spatio-temporal signatures have diversity on similarity score measurement and parameter dependence. Hence, we have distinguished these two schemes for similarity comparisons to improve the understandability.

Spatial Feature; As we mentioned in Chapter 2, four visual descriptor methods have been utilized in the scope of this thesis including SIFT, Opponent SIFT, Flip-SIFT and SURF on sparsely sampled Hessian-Laplace interest points. Since all these descriptors are scale and

orientation invariant, each descriptor has own scale and orientation parameters alongside of spatial coordinates. Additionally, a visual group binary signature is encoded to improve the matching accuracy. Thus, for each entry per descriptor, extra parameters are inserted as;

- the visual group binary signature b_{vg},
- the uniformly quantized orientation q_θ,
- the logarithmically quantized scale q_s,
- the coordinate of interest point in spatial domain (x, y),

In this scheme, the similarity score between two frames depends on similarity of visual indexing $s_{model}(\boldsymbol{v}^{t_i,m}, \boldsymbol{v}^{t_j,n})$, similarity of geometric signature $s_{vg}({b_{vg}}^{t_i,m}, {b_{vg}}^{t_j,n})$, term frequency of reference descriptor ${w_{tfidf}}^{t_i,m}$ and geometric consistency post-processing filtering as shown in Figure 1.

Therefore, we have combined the visual indexing (bag-of-word (*BoW*), hamming embedding (*HE*), product quantization (*PQ*)), geometric signature (visual group binary signature (*VGBS*)) and geometric consistency (weak geometric consistency with scale difference (*SWGC*) and weak geometric consistency with translation difference (*TWGC*)) methods with each feature model to investigate the effects on performance.

Spatio-Temporal Feature; For this descriptor type, we have mentioned three descriptor models that exploit visual or temporal content of consecutive frames including HoG and MBH. Additionally, these descriptors are sampled from dense trajectories. Similar to spatial descriptor, there are several parameters that are specific for this feature model. Thus, these parameters are inserted on each entry as;

- the uniformly quantized scale q_s,
- the means of coordinates of trajectory points (μ_x, μ_y),
- the variances of coordinates of trajectory points (σ_x, σ_y),

Since the geometric signature is not encoded in this domain, the similarity score depends on similarity of visual indexing $s_{model}(\boldsymbol{v}^{t_i,m}, \boldsymbol{v}^{t_j,n})$, term frequency of reference descriptor ${w_{tfidf}}^{t_i,m}$ and geometric consistency filtering post-processing stage. Therefore we will examine the effects of combination of all indexing methods and feature extraction with spatio-temporal geometric consistency (*STWGC*) as expressed in Figure 2.

5.2.2 Experiments

In this section, we will discuss the evaluation results on different combination of feature extraction, indexing and geometric verification schemes in terms of recall, precision and f1-scores in addition to memory usage and comparison time.

For that purpose, recall, precision and f1-scores are presented in Table 3-20. These scores are obtained for top retrieved reference video. From the results, for both spatial and spatio-temporal descriptors, hamming embedding and product quantization schemes yield overwhelmingly better results over classical bag-of-word representation. The main reason is that these methods take in-class location of the vector into account with an additional code.

Second observation is that employing soft similarity score assignment on hamming embedding and product quantization contribute the detection accuracy. Especially, these assumption improve the performance on the attacks that deform the vector structure like re-encoding and white noise. Hence, we can say that distinctive characteristic of video copy detection from duplicate video search is that it should be robust against large variation within cluster owing to possibility of dealing with noisy version of features.

Another observation is that, despite opponent SIFT descriptor proves that it is more discriminative over classical SIFT and SURF descriptors in many computer vision tasks [26], it fails and gives worse results on this domain. The underlying reason is that in quantization stage, the descriptors with different length are mapped into same cluster center size and bit

rate per component of vector becomes higher for the descriptor whose feature size is smaller. Thus, particularly for this task, extending the representation dimension is not a good solution even if discriminative power of descriptor is increased.

Owing to invariance against scale, orientation and flip, Flip Invariant SIFT (F-SIFT) gives the best overall performance for all recall, precision and f1 score metrics. However, one important observation can be made with SIFT descriptor. The performance on re-encoding attacks is decreased in F-SIFT descriptor compared to SIFT descriptor. The possible reason is that curl computation which determines the local region must be flipped or not mislead on re-encoding attacks.

Representing video purely with the spatio-temporal signatures that are not static gives worse performance over spatial signatures. The core reason is for some case, the background can have distinctive information. Therefore, joint usage of spatio and spatio-temporal signatures would yield better performance besides increasing computational complexity.

The main observation on spatio-temporal signature is that exploiting motion content of local regions particularly fails on re-encoding attacks owing to smoothing and windowing (during re-encoding) deformations deform optical flow field.

The filtering stage by utilizing geometric consistency among local signatures for both spatial and spatio-temporal improves the performance significantly. Especially, leveraging translation difference in consistency outperforms scale difference. Additionally, for spatio-temporal domain, insertion of spatial variation constraint filters out outliers at the beginning of stage.

To illustrate the recall performance on overall top 10 score, we give Figures 12-17. From the figures, our developed system reaches to 0.8888 recall accuracy in top 10 results with the combination of flip invariant SIFT, hamming embedding and geometric consistency with translation difference.

Table 1. Comparison time for spatial descriptor models.

Feature Model	BoW+SWGC	BoW+TWGC	BoW+SWGC +VGBS	BoW+TWGC +VGBS	HE+SWGC	HE+TWGC	HE+SWGC +VGBS	HE+TWGC +VGBS	PQ+SWGC	PQ+TWGC	PQ+SWGC +VGBS	BoW+TW +VGBS
Spatial	0.04 sn	0.1 sn	0.06 sn	0.12 sn	0.05 sn	0.11 sn	0.07 sn	0.13 sn	0.07 sn	0.12 sn	0.09 sn	0.14 sn

Table 2. Comparison time for spatio-temporal descriptor models.

Feature Model	BoW	BoW+STWGC	HE	HE+STWGC	PQ	PQ+STWGC
Spatio-Temporal	0.01 sn	0.05 sn	0.02 sn	0.07 sn	0.03 sn	0.09 sn

Due to the fact that each visual descriptor model follows the similar procedure on feature extraction and quantization-based indexing, the memory requirement would not alter according to descriptor model (Similarly, for spatio-temporal descriptors). According to memory usage, the developed system can work seamlessly on a laptop without causing any trouble.

Even though signature comparison depends on the distribution of visual codeword which is related to the discriminative power of descriptor, mainly, this amount is negligible. Thus, similar to memory usage, estimation of comparison time on indexing and geometric verification stages makes more sense. In Table 1 and Table 2, for both spatial and spatio-temporal descriptors, comparison time is given in second for comparing 1 second query video with 100 hours of reference database.

From the comparison time results, if we accept bag-of-word representation as a baseline, hamming embedding and product quantization schemes increase the comparison time owing to counting in-class locality information. Adding geometric consistency for each feature domain also causes an increase. Particularly, geometric consistency with translation (*TWGC*) boosts the time compare to geometric consistency with scale (*SWGC*).

Interestingly, combination of visual group binary signature and geometric consistency with scale difference yields compatible performance results over more complex method like geometric consistency with translation with smaller comparison time.

Table 3. Recall scores for SIFT descriptor.

Baseline	T2	T3	T4	T5	T6	T8	T10	Overall
BoW+SWGC	0.4328	0.7686	0.6044	0.8283	0.8955	0.3283	0.3582	0.6023
BoW+TWGC	0.6343	0.8955	0.8208	0.9104	0.9701	0.4179	0.4925	0.7345
BoW+SWGC+VGBS	0.5074	0.8805	0.7238	0.9029	0.9626	0.3507	0.4179	0.6780
BoW+TWGC+VGBS	0.6044	0.9179	0.8283	0.9328	0.9776	0.4402	0.4925	0.7420
HE+SWGC	0.6417	0.8955	0.7611	0.9179	0.9701	0.3880	0.4701	0.7206
HE+TWGC	0.7313	0.9402	0.8731	0.9626	1.0	0.5	0.5522	0.7942
HE+SWGC+VGBS	0.6865	0.9328	0.8358	0.9477	0.9925	0.4402	0.5223	0.7654
HE+TWGC+VGBS	0.7462	0.9701	0.8955	0.9552	1.0	0.4850	0.5522	0.8006
PQ+SWGC	0.6641	0.9253	0.8059	0.9104	0.9701	0.4029	0.4104	0.7270
PQ+TWGC	0.7164	0.9402	0.8656	0.9402	0.9850	0.4626	0.5074	0.7739
PQ+SWGC+VGBS	0.6865	0.9328	0.8507	0.9402	0.9850	0.4326	0.4626	0.7558
PQ+TWGC+VGBS	0.6865	0.9701	0.9029	0.9477	0.9850	0.4626	0.5298	0.7858

Table 4. Precision scores for SIFT descriptor.

Baseline	T2	T3	T4	T5	T6	T8	T10	Overall
BoW+SWGC	1.0	1.0	0.9878	0.9910	0.9917	1.0	1.0	0.9947
BoW+TWGC	1.0	1.0	0.9909	0.9918	0.9923	1.0	1.0	0.9956
BoW+SWGC+VGBS	1.0	1.0	0.9897	0.9918	0.9923	1.0	1.0	0.9953
BoW+TWGC+VGBS	1.0	1.0	0.9910	0.9920	0.9924	1.0	1.0	0.9957
HE+SWGC	1.0	1.0	0.9902	0.9919	0.9923	1.0	1.0	0.9955
HE+TWGC	1.0	1.0	0.9915	0.9923	0.9925	1.0	1.0	0.9959
HE+SWGC+VGBS	1.0	1.0	0.9911	0.9921	0.9925	1.0	1.0	0.9958
HE+TWGC+VGBS	1.0	1.0	0.9917	0.9922	0.9925	1.0	1.0	0.9960
PQ+SWGC	1.0	1.0	0.9908	0.9918	0.9923	1.0	1.0	0.9956
PQ+TWGC	1.0	1.0	0.9914	0.9921	0.9924	1.0	1.0	0.9958
PQ+SWGC+VGBS	1.0	1.0	0.9913	0.9921	0.9924	1.0	1.0	0.9957
PQ+TWGC+VGBS	1.0	1.0	0.9918	0.9921	0.9924	1.0	1.0	0.9959

Table 5. F1 scores for SIFT descriptor.

Baseline	T2	T3	T4	T5	T6	T8	T10	Overall
BoW+SWGC	0.6041	0.8691	0.75	0.9024	0.9411	0.4943	0.5274	0.7503
BoW+TWGC	0.7762	0.9448	0.8979	0.9494	0.9811	0.5894	0.6666	0.8453
BoW+SWGC+VGBS	0.6732	0.9365	0.8362	0.9453	0.9772	0.5193	0.5894	0.8065
BoW+TWGC+VGBS	0.7534	0.9571	0.9024	0.9615	0.9849	0.6113	0.6666	0.8503
HE+SWGC	0.7818	0.9448	0.8607	0.9534	0.9811	0.5591	0.6395	0.8361
HE+TWGC	0.8448	0.9692	0.9285	0.9772	0.9962	0.6666	0.7115	0.8837
HE+SWGC+VGBS	0.8141	0.9652	0.9068	0.9694	0.9924	0.6113	0.6862	0.8655
HE+TWGC+VGBS	0.8547	0.9848	0.9411	0.9733	0.9962	0.6532	0.7115	0.8877
PQ+SWGC	0.7982	0.9612	0.8888	0.9494	0.9811	0.5744	0.5820	0.8404
PQ+TWGC	0.8347	0.9692	0.9243	0.9655	0.9887	0.6326	0.6732	0.8710
PQ+SWGC+VGBS	0.8141	0.9652	0.9156	0.9655	0.9887	0.6041	0.6326	0.8593
PQ+TWGC+VGBS	0.8141	0.9848	0.9453	0.9694	0.9887	0.6326	0.6926	0.8770

Table 6. Recall scores for Opponent SIFT descriptor.

Baseline	T2	T3	T4	T5	T6	T8	T10	Overall
BoW+SWGC	0.3582	0.7164	0.4328	0.7164	0.8432	0.2761	0.2388	0.5117
BoW+TWGC	0.5820	0.8432	0.7611	0.8731	0.9552	0.4104	0.4552	0.6972
BoW+SWGC+VGBS	0.5	0.8059	0.6194	0.8059	0.9104	0.3656	0.4104	0.6311
BoW+TWGC+VGBS	0.5895	0.8805	0.7910	0.9104	0.9626	0.4253	0.4776	0.7196
HE+SWGC	0.5820	0.8059	0.5820	0.8208	0.9104	0.3656	0.3955	0.6375
HE+TWGC	0.6940	0.8880	0.8283	0.8955	0.9925	0.4477	0.5223	0.7526
HE+SWGC+VGBS	0.6641	0.8731	0.7611	0.8507	0.9701	0.3731	0.5074	0.7142
HE+TWGC+VGBS	0.6940	0.8955	0.8358	0.9029	0.9925	0.4552	0.5223	0.7569
PQ+SWGC	0.5895	0.8134	0.5	0.7985	0.8955	0.3731	0.3358	0.6151
PQ+TWGC	0.6641	0.8656	0.7238	0.8731	0.9626	0.4253	0.4552	0.7100
PQ+SWGC+VGBS	0.6567	0.8731	0.6492	0.8432	0.9402	0.4402	0.4328	0.6908
PQ+TWGC+VGBS	0.6641	0.8880	0.7388	0.8582	0.9626	0.4402	0.4701	0.7174

Table 7. Precision scores for Opponent SIFT descriptor.

Baseline	T2	T3	T4	T5	T6	T8	T10	Overall
BoW+SWGC	1.0	1.0	0.9830	0.9896	0.9912	1.0	1.0	0.9937
BoW+TWGC	1.0	1.0	0.9902	0.9915	0.9922	1.0	1.0	0.9954
BoW+SWGC+VGBS	1.0	1.0	0.9880	0.9908	0.9918	1.0	1.0	0.9949
BoW+TWGC+VGBS	1.0	1.0	0.9906	0.9918	0.9923	1.0	1.0	0.9955
HE+SWGC	1.0	1.0	0.9873	0.9909	0.9918	1.0	1.0	0.9950
HE+TWGC	1.0	1.0	0.9910	0.9917	0.9925	1.0	1.0	0.9957
HE+SWGC+VGBS	1.0	1.0	0.9902	0.9913	0.9923	1.0	1.0	0.9962
HE+TWGC+VGBS	1.0	1.0	0.9911	0.9918	0.9925	1.0	1.0	0.9964
PQ+SWGC	1.0	1.0	0.9852	0.9907	0.9917	1.0	1.0	0.9948
PQ+TWGC	1.0	1.0	0.9897	0.9915	0.9923	1.0	1.0	0.9955
PQ+SWGC+VGBS	1.0	1.0	0.9886	0.9912	0.9921	1.0	1.0	0.9953
PQ+TWGC+VGBS	1.0	1.0	0.99	0.9913	0.9923	1.0	1.0	0.9955

Table 8. F1 scores for Opponent SIFT descriptor.

Baseline	T2	T3	T4	T5	T6	T8	T10	Overall
BoW+SWGC	0.5274	0.8347	0.6010	0.8311	0.9112	0.4327	0.3854	0.6755
BoW+TWGC	0.7358	0.9149	0.8607	0.9285	0.9733	0.5820	0.6256	0.8200
BoW+SWGC+VGBS	0.6666	0.8925	0.7614	0.8888	0.9494	0.5355	0.5820	0.7723
BoW+TWGC+VGBS	0.7417	0.9365	0.8796	0.9494	0.9772	0.5968	0.6464	0.8353
HE+SWGC	0.7358	0.8925	0.7323	0.8979	0.9494	0.5355	0.5668	0.7771
HE+TWGC	0.8193	0.9407	0.9024	0.9411	0.9925	0.6185	0.6862	0.8573
HE+SWGC+VGBS	0.7982	0.9322	0.8607	0.9156	0.9811	0.5434	0.6732	0.8320
HE+TWGC+VGBS	0.8193	0.9449	0.9068	0.9453	0.9925	0.6256	0.6862	0.8603
PQ+SWGC	0.7417	0.8971	0.6633	0.8842	0.9411	0.5434	0.5027	0.7602
PQ+TWGC	0.7920	0.928	0.8362	0.9285	0.9772	0.5968	0.6256	0.8288
PQ+SWGC+VGBS	0.7927	0.9322	0.7837	0.9112	0.9655	0.6113	0.6041	0.8156
PQ+TWGC+VGBS	0.7982	0.9407	0.8461	0.92	0.9772	0.6113	0.6395	0.8339

Table 9. Recall scores for F-SIFT descriptor.

Baseline	T2	T3	T4	T5	T6	T8	T10	Overall
BoW+SWGC	0.4029	0.7761	0.5373	0.8208	0.8731	0.4552	0.3731	0.6055
BoW+TWGC	0.6044	0.8955	0.7910	0.9253	0.9701	0.7248	0.6119	0.7889
BoW+SWGC+VGBS	0.4850	0.8805	0.6940	0.8731	0.9477	0.5298	0.4925	0.7004
BoW+TWGC+VGBS	0.6044	0.9253	0.7985	0.9402	0.9776	0.7248	0.6194	0.7985
HE+SWGC	0.5970	0.8880	0.7238	0.9029	0.9552	0.5970	0.5597	0.7462
HE+TWGC	0.7313	0.9328	0.8582	0.9552	0.9850	0.8059	0.6791	0.8496
HE+SWGC+VGBS	0.6791	0.9402	0.8059	0.9477	0.9850	0.7014	0.6343	0.8134
HE+TWGC+VGBS	0.7313	0.9552	0.8432	0.9552	0.9850	0.8059	0.6865	0.8516
PQ+SWGC	0.6417	0.9104	0.7388	0.9029	0.9477	0.5820	0.5149	0.7484
PQ+TWGC	0.6940	0.9328	0.8432	0.9477	0.9776	0.7611	0.6492	0.8294
PQ+SWGC+VGBS	0.6791	0.9253	0.7910	0.9253	0.9701	0.6343	0.5671	0.7846
PQ+TWGC+VGBS	0.7014	0.9328	0.8358	0.9552	0.9776	0.7761	0.6492	0.8326

Table 10. Precision scores for F-SIFT descriptor.

Baseline	T2	T3	T4	T5	T6	T8	T10	Overall
BoW+SWGC	1.0	1.0	0.9863	0.9909	0.9915	1.0	1.0	0.9947
BoW+TWGC	1.0	1.0	0.9906	0.992	0.9923	1.0	1.0	0.9959
BoW+SWGC+VGBS	1.0	1.0	0.9893	0.9915	0.9921	1.0	1.0	0.9954
BoW+TWGC+VGBS	1.0	1.0	0.9907	0.9921	0.9924	1.0	1.0	0.9959
HE+SWGC	1.0	1.0	0.9897	0.9918	0.9922	1.0	1.0	0.9957
HE+TWGC	1.0	1.0	0.9913	0.9922	0.9924	1.0	1.0	0.9962
HE+SWGC+VGBS	1.0	1.0	0.9908	0.9921	0.9924	1.0	1.0	0.9960
HE+TWGC+VGBS	1.0	1.0	0.9912	0.9922	0.9924	1.0	1.0	0.9962
PQ+SWGC	1.0	1.0	0.99	0.9918	0.9921	1.0	1.0	0.9957
PQ+TWGC	1.0	1.0	0.9912	0.9921	0.9924	1.0	1.0	0.9961
PQ+SWGC+VGBS	1.0	1.0	0.9906	0.992	0.9923	1.0	1.0	0.9959
PQ+TWGC+VGBS	1.0	1.0	0.9911	0.9922	0.9924	1.0	1.0	0.9961

Table 11. F1 scores for F-SIFT descriptor.

Baseline	T2	T3	T4	T5	T6	T8	T10	Overall
BoW+SWGC	0.5744	0.8739	0.6956	0.8979	0.9285	0.6256	0.5434	0.7581
BoW+TWGC	0.7534	0.9444	0.8796	0.9575	0.9811	0.8398	0.7592	0.8804
BoW+SWGC+VGBS	0.6532	0.9365	0.8157	0.9285	0.9694	0.6926	0.66	0.8222
BoW+TWGC+VGBS	0.7534	0.9612	0.8842	0.9655	0.9849	0.8404	0.7649	0.8833
HE+SWGC	0.7476	0.9407	0.8362	0.9453	0.9733	0.7476	0.7177	0.8531
HE+TWGC	0.8448	0.9652	0.92	0.9733	0.9887	0.8925	0.8088	0.9171
HE+SWGC+VGBS	0.8088	0.9692	0.8888	0.9694	0.9887	0.8245	0.7762	0.8955
HE+TWGC+VGBS	0.8448	0.9770	0.9112	0.9733	0.9887	0.8925	0.8141	0.9182
PQ+SWGC	0.7818	0.9531	0.8461	0.9453	0.9694	0.7358	0.6798	0.8545
PQ+TWGC	0.8193	0.9652	0.9112	0.9694	0.9849	0.8644	0.7873	0.9051
PQ+SWGC+VGBS	0.8088	0.9612	0.8796	0.9575	0.9811	0.7762	0.7238	0.8777
PQ+TWGC+VGBS	0.8245	0.9652	0.9068	0.9733	0.9849	0.8739	0.7872	0.9070

Table 12. Recall scores for SURF descriptor.

Baseline	T2	T3	T4	T5	T6	T8	T10	Overall
BoW+SWGC	0.5	0.8358	0.6044	0.8432	0.9253	0.3507	0.3283	0.6268
BoW+TWGC	0.5970	0.9029	0.8134	0.9253	0.9626	0.4179	0.4552	0.7249
BoW+SWGC+VGBS	0.5746	0.9104	0.7089	0.8955	0.9626	0.3880	0.4328	0.6961
BoW+TWGC+VGBS	0.5895	0.9104	0.8283	0.9328	0.9701	0.4626	0.4701	0.7377
HE+SWGC	0.6865	0.9104	0.7761	0.9253	0.9626	0.4104	0.4552	0.7324
HE+TWGC	0.7164	0.9402	0.8731	0.9552	0.9850	0.4925	0.5	0.7803
HE+SWGC+VGBS	0.7238	0.9552	0.7910	0.9522	0.9776	0.4552	0.4776	0.7622
HE+TWGC+VGBS	0.6940	0.9402	0.8507	0.9328	0.9850	0.4477	0.5074	0.7654
PQ+SWGC	0.5476	0.8656	0.6268	0.8432	0.9477	0.3731	0.3805	0.6545
PQ+TWGC	0.6044	0.9029	0.8208	0.9477	0.9626	0.4477	0.4626	0.7356
PQ+SWGC+VGBS	0.5746	0.9104	0.7089	0.9253	0.9552	0.4179	0.4104	0.7004
PQ+TWGC+VGBS	0.5970	0.9253	0.8358	0.9402	0.9850	0.4477	0.4701	0.7430

Table 13. Precision scores for SURF descriptor.

Baseline	T2	T3	T4	T5	T6	T8	T10	Overall
BoW+SWGC	1.0	1.0	0.9878	0.9912	0.992	1.0	1.0	0.9949
BoW+TWGC	1.0	1.0	0.9909	0.992	0.9923	1.0	1.0	0.9956
BoW+SWGC+VGBS	1.0	1.0	0.9895	0.9917	0.9923	1.0	1.0	0.9954
BoW+TWGC+VGBS	1.0	1.0	0.9910	0.9920	0.9923	1.0	1.0	0.9956
HE+SWGC	1.0	1.0	0.9904	0.992	0.9923	1.0	1.0	0.9956
HE+TWGC	1.0	1.0	0.9915	0.9922	0.9924	1.0	1.0	0.9959
HE+SWGC+VGBS	1.0	1.0	0.9906	0.9922	0.9924	1.0	1.0	0.9958
HE+TWGC+VGBS	1.0	1.0	0.9913	0.9920	0.9924	1.0	1.0	0.9958
PQ+SWGC	1.0	1.0	0.9882	0.9921	0.9921	1.0	1.0	0.9951
PQ+TWGC	1.0	1.0	0.9909	0.9921	0.9923	1.0	1.0	0.9956
PQ+SWGC+VGBS	1.0	1.0	0.9895	0.992	0.9922	1.0	1.0	0.9954
PQ+TWGC+VGBS	1.0	1.0	0.9911	0.9921	0.9924	1.0	1.0	0.9957

Table 14. F1 scores for SURF descriptor.

Baseline	T2	T3	T4	T5	T6	T8	T10	Overall
BoW+SWGC	0.6666	0.9105	0.75	0.9912	0.9575	0.5193	0.4943	0.7691
BoW+TWGC	0.7476	0.9490	0.8934	0.9575	0.9772	0.5894	0.6256	0.8389
BoW+SWGC+VGBS	0.7298	0.9531	0.8260	0.9411	0.9772	0.5591	0.6041	0.8193
BoW+TWGC+VGBS	0.7417	0.9531	0.9024	0.9615	0.9811	0.6326	0.6395	0.8475
HE+SWGC	0.8141	0.9531	0.8702	0.9575	0.9772	0.5820	0.6256	0.8439
HE+TWGC	0.8347	0.9692	0.9285	0.9733	0.9887	0.66	0.6666	0.8750
HE+SWGC+VGBS	0.8398	0.9770	0.8796	0.9733	0.9849	0.6256	0.6464	0.8635
HE+TWGC+VGBS	0.8193	0.9692	0.9156	0.9615	0.9887	0.6185	0.6732	0.8655
PQ+SWGC	0.7053	0.928	0.7612	0.9112	0.9694	0.5434	0.5513	0.7897
PQ+TWGC	0.7534	0.9490	0.8979	0.9694	0.9772	0.6185	0.6326	0.8461
PQ+SWGC+VGBS	0.7298	0.9531	0.8260	0.9575	0.9733	0.5894	0.5820	0.8222
PQ+TWGC+VGBS	0.7476	0.9612	0.9068	0.9655	0.9887	0.6185	0.6395	0.8510

Table 15. Recall scores for HoG trajectory descriptor.

Baseline	T2	T3	T4	T5	T6	T8	T10	Overall
BoW	0.1641	0.6194	0.3432	0.5671	0.3134	0.2238	0.1716	0.3432
BoW+STWGC	0.3208	0.8134	0.5895	0.7910	0.7014	0.3432	0.3283	0.5554
HE	0.2313	0.7164	0.4253	0.7089	0.4776	0.3358	0.2388	0.4476
HE+STWGC	0.3208	0.8507	0.6268	0.8358	0.7910	0.3805	0.3880	0.5991
PQ	0.2463	0.7164	0.4020	0.6940	0.4253	0.3432	0.2164	0.4349
PQ+STWGC	0.3359	0.8657	0.6119	0.8507	0.7761	0.3881	0.3880	0.6023

Table 16. Precision scores for HoG trajectory descriptor.

Baseline	T2	T3	T4	T5	T6	T8	T10	Overall
BoW	1.0	0.9880	1.0	0.9870	1.0	0.9677	1.0	0.9907
BoW+STWGC	1.0	0.9909	1.0	0.9906	1.0	0.9787	1.0	0.9942
HE	1.0	0.9896	1.0	0.9895	1.0	0.9782	1.0	0.9929
HE+STWGC	1.0	0.9913	1.0	0.9915	1.0	0.9807	1.0	0.9946
PQ	1.0	0.9869	1.0	0.9864	1.0	0.9787	1.0	0.9939
PQ+STWGC	1.0	0.9915	1.0	0.9913	1.0	0.9811	1.0	0.9948

Table 17. F1 scores for HoG trajectory descriptor.

Baseline	T2	T3	T4	T5	T6	T8	T10	Overall
BoW	0.2820	0.7614	0.5111	0.7203	0.4772	0.3636	0.2929	0.5098
BoW+STWGC	0.4858	0.8934	0.7417	0.8796	0.8245	0.5082	0.4943	0.7127
HE	0.3757	0.8311	0.5968	0.8260	0.6464	0.5	0.3855	0.6171
HE+STWGC	0.4858	0.9156	0.7706	0.9068	0.8833	0.5483	0.5591	0.7478
PQ	0.3952	0.8311	0.5744	0.8157	0.5968	0.5082	0.3558	0.6050
PQ+STWGC	0.5028	0.9243	0.7592	0.9156	0.8739	0.5561	0.5591	0.7503

Table 18. Recall scores for MBH trajectory descriptor.

Baseline	T2	T3	T4	T5	T6	T8	T10	Overall
BoW	0.044	0.3507	0	0.3134	0.1716	0.0970	0.0522	0.1471
BoW+STWGC	0.1492	0.5820	0.0671	0.5970	0.5447	0.2089	0.1268	0.3251
HE	0.1044	0.4104	0	0.3880	0.2388	0.1492	0.1044	0.1993
HE+STWGC	0.1417	0.6044	0.0895	0.6343	0.6119	0.2313	0.1641	0.3539
PQ	0.1119	0.3955	0	0.3955	0.2239	0.1492	0.1119	0.1982
PQ+STWGC	0.1343	0.5970	0.0895	0.6418	0.5970	0.2238	0.1716	0.3507

Table 19. Precision scores for MBH trajectory descriptor.

Baseline	T2	T3	T4	T5	T6	T8	T10	Overall
BoW	1.0	0.9791	nan	0.9767	1.0	0.9285	1.0	0.9787
BoW+STWGC	1.0	0.9873	1.0	0.9876	1.0	0.9655	1.0	0.9902
HE	1.0	0.9821	nan	0.9811	1.0	0.9523	1.0	0.9842
HE+STWGC	1.0	0.9878	1.0	0.9883	1.0	0.9687	1.0	0.9910
PQ	1.0	0.9815	nan	0.9515	1.0	0.9524	1.0	0.9841
PQ+STWGC	1.0	0.9877	1.0	0.9885	1.0	0.9677	1.0	0.9919

Table 20. F1 scores for MBH trajectory descriptor.

Baseline	T2	T3	T4	T5	T6	T8	T10	Overall
BoW	0.0857	0.5164	nan	0.4745	0.2929	0.1756	0.0992	0.2557
BoW+STWGC	0.2597	0.7323	0.1258	0.7441	0.7053	0.3435	0.2251	0.4895
HE	0.1891	0.5789	nan	0.5561	0.3855	0.2580	0.1891	0.3315
HE+STWGC	0.2483	0.75	0.1643	0.7727	0.7592	0.3734	0.2820	0.5216
PQ	0.2013	0.5638	nan	0.5587	0.3658	0.2580	0.2013	0.3299
PQ+STWGC	0.2367	0.7441	0.1643	0.7782	0.7476	0.3636	0.2929	0.5182

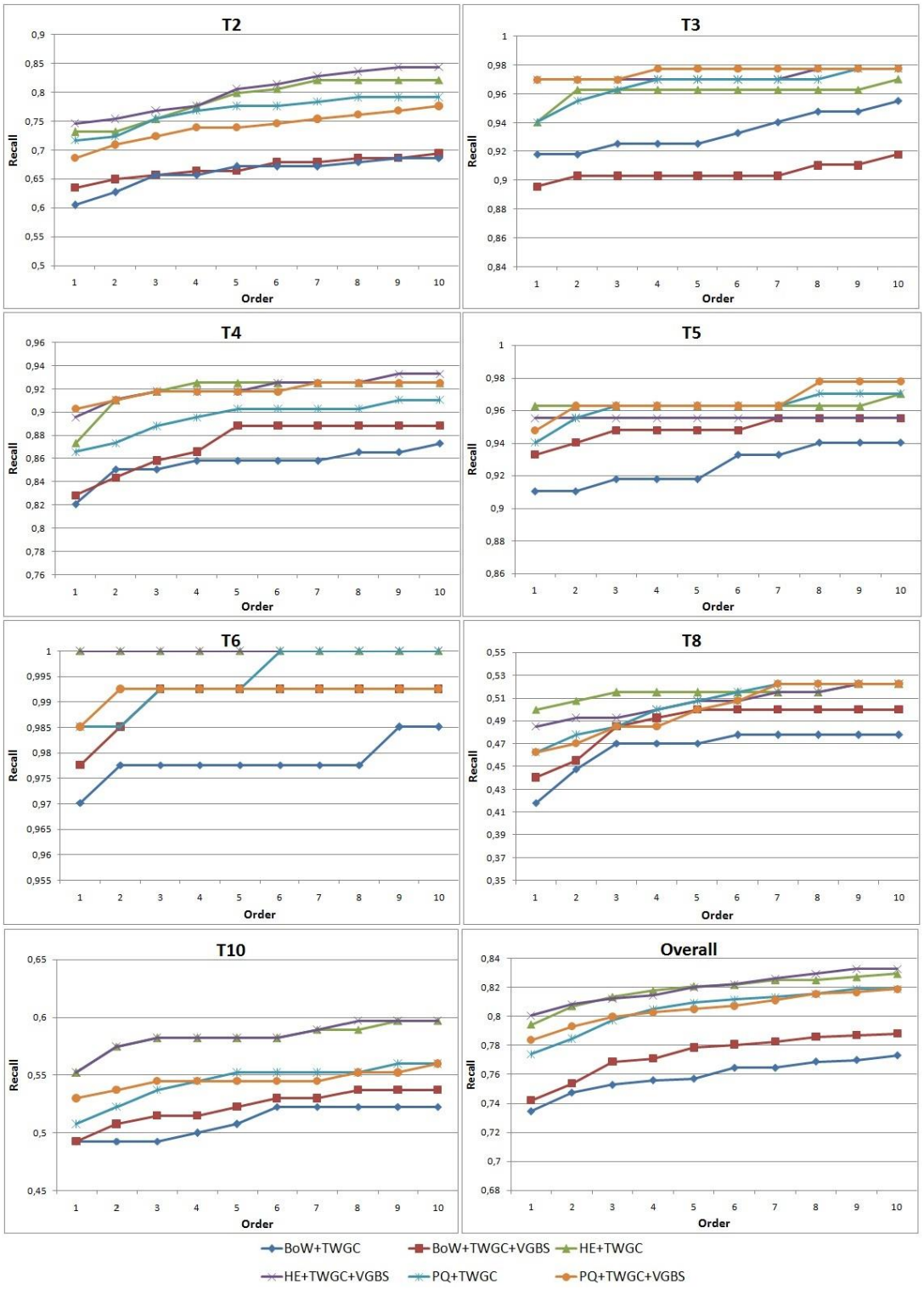

Figure 12. Top 10 recall scores for SIFT descriptor.

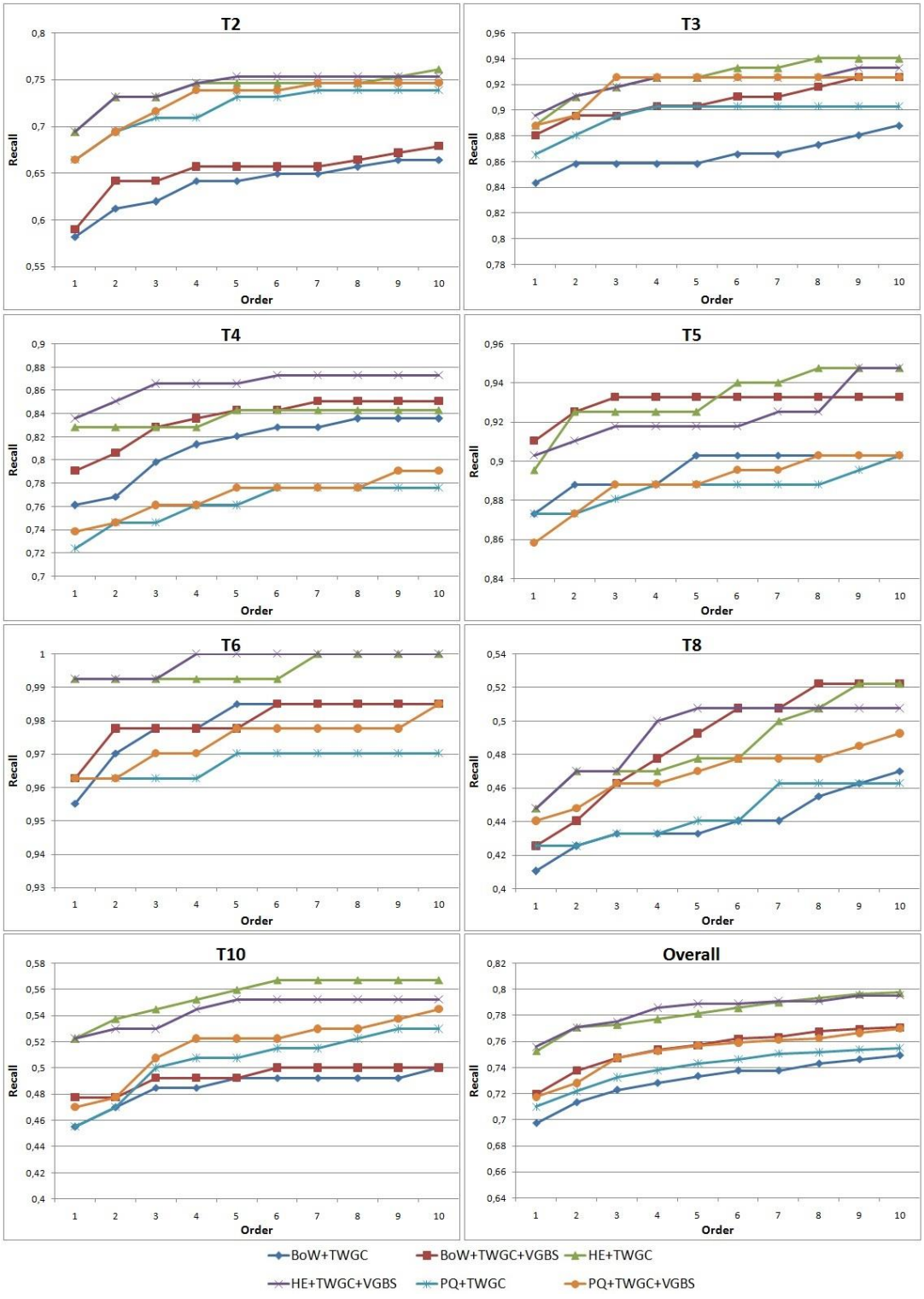

Figure 13. Top 10 recall scores for Opponent SIFT descriptor.

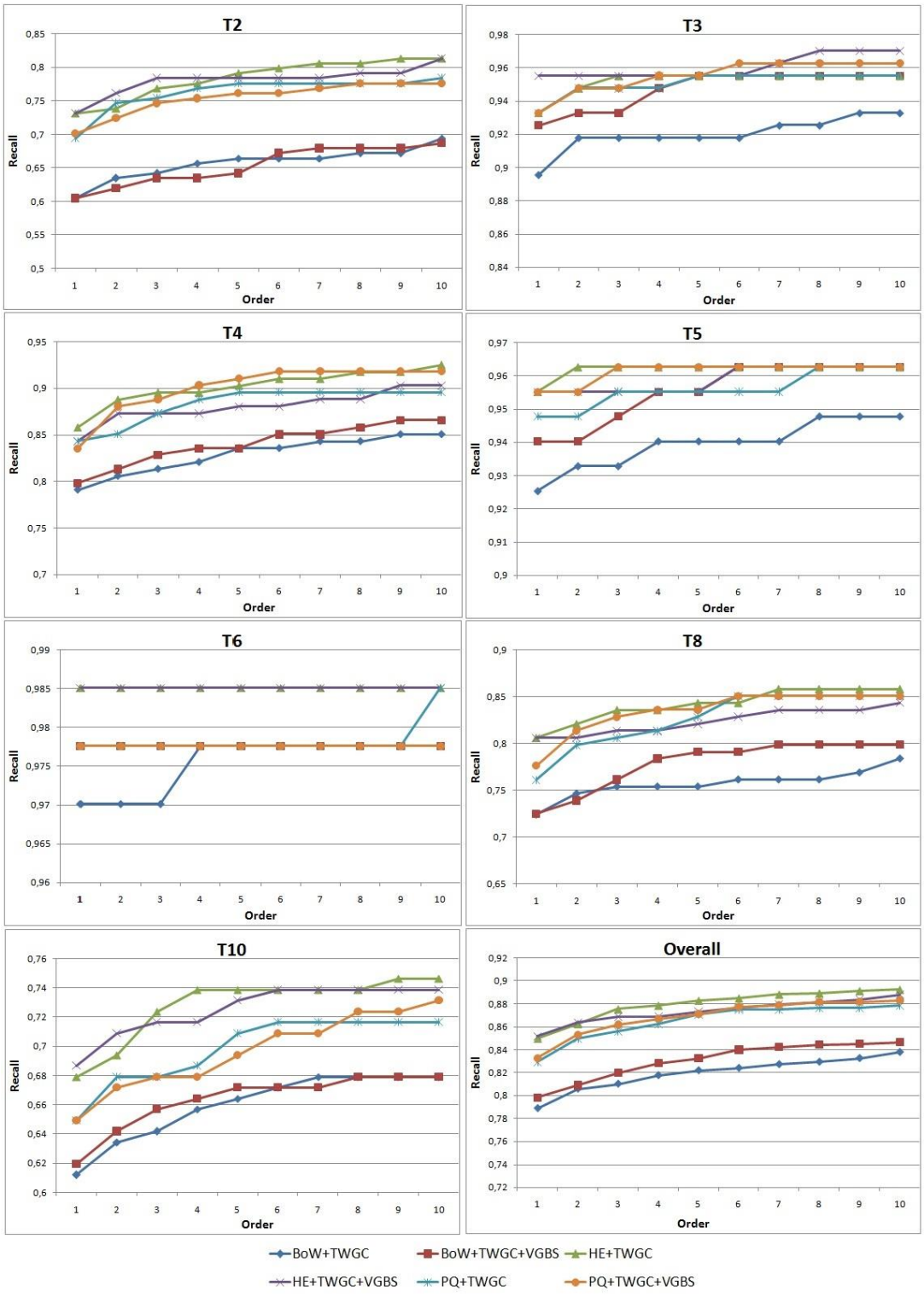

Figure 14. Top 10 recall scores for F-SIFT descriptor.

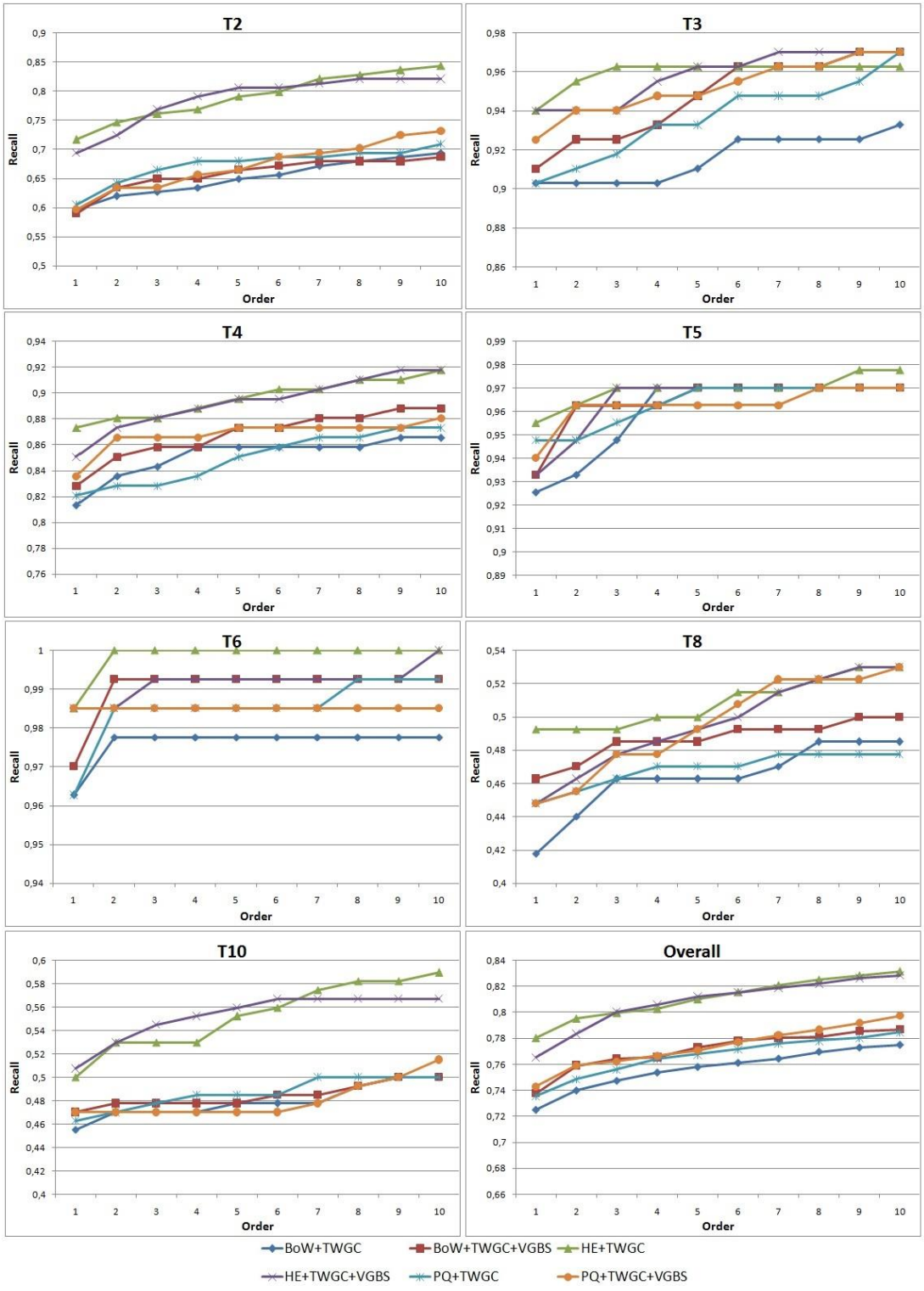

Figure 15. Top 10 recall scores for SURF descriptor.

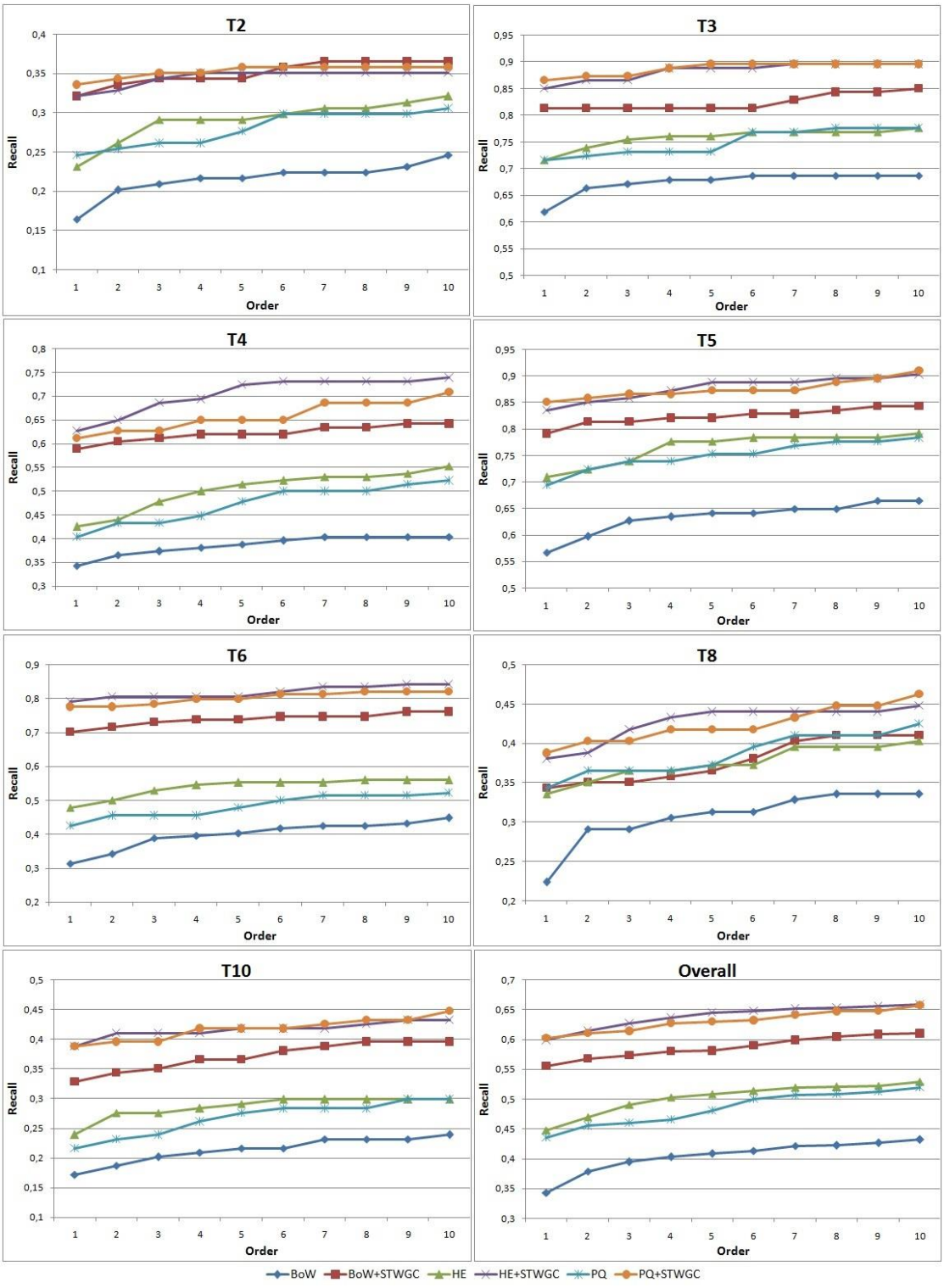

Figure 16. Top 10 recall scores for HoG trajectory descriptor.

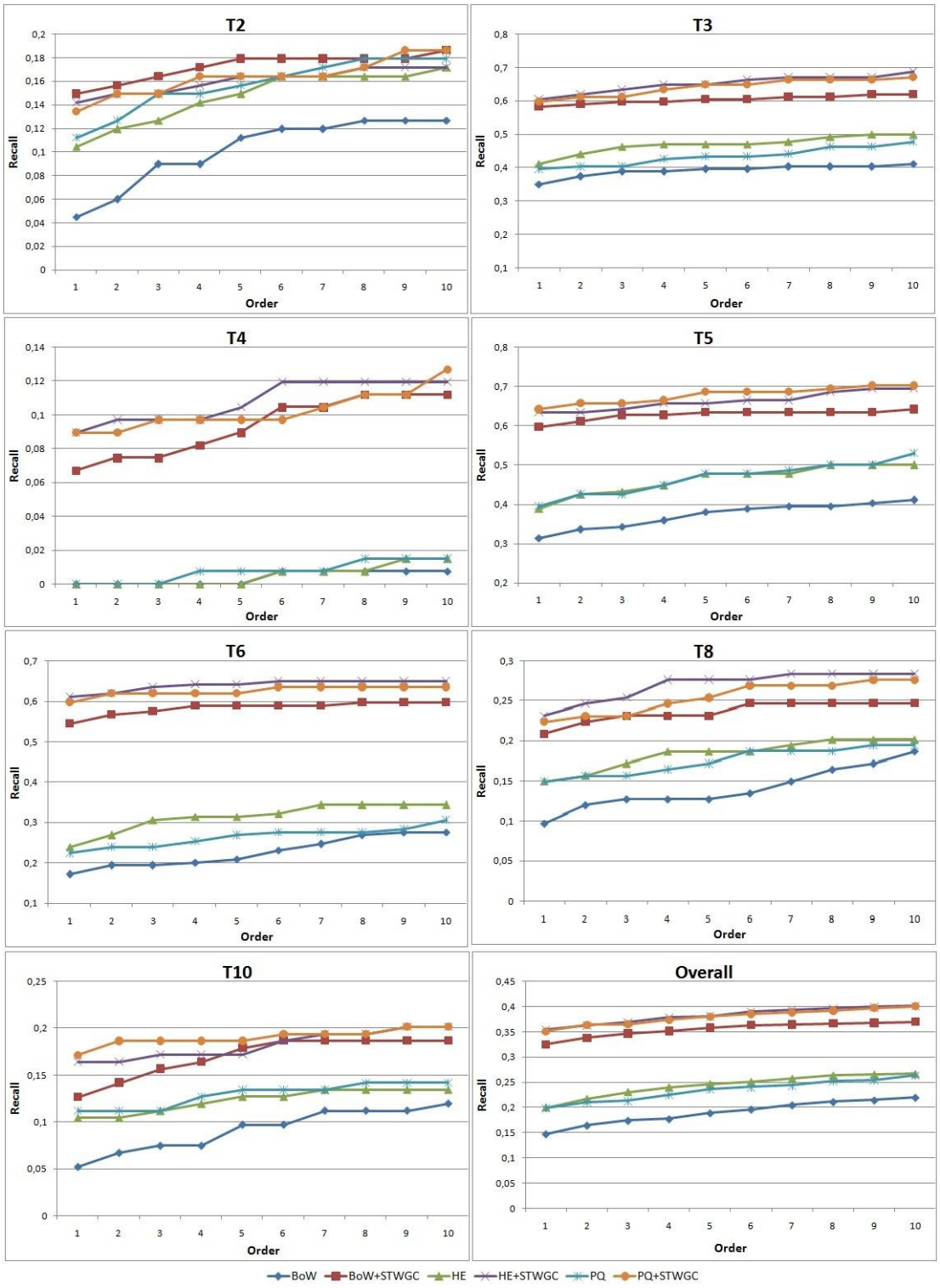

Figure 17. Top 10 recall scores for MBH trajectory descriptor.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusions

In this thesis, we propose an overall content-based copy detection system. It has three main stages namely feature extraction, quantization-based indexing and geometric verification. The performance of developed system is assessed by recall, precision and f1-score performance metrics besides memory and comparison time. From the results, combination of flip invariant version of scale invariant feature transform (F-SIFT), hamming embedding, geometric consistency with translation difference and visual group binary signature yield most feasible performance.

Within the scope of this thesis, we have developed several novel solutions on content-based copy detection. In feature extraction stage, we have primarily deployed dense trajectory feature models on this task which is initially proposed for action recognition.

We have observed that hardcoded similarity comparison reduces the performance on copy detection. Thus, we have utilized soft assignment score metrics for hamming and product quantization-based content indexing schemes.

In order to constitute local geometric relation on local features, we have proposed a novel signature that is compact and effective at the same time. The result shows that encoding neighboring relation with a binary signature gives overwhelmingly good performances besides ease of applicability on large datasets.

In geometric consistency stage, leveraging similar idea of weak geometric consistency in spatial domain, we have proposed a novel spatio-temporal weak geometric consistency stage.

This method exploits the feature characteristics of spatio-temporal signatures in spatial and time domain simultaneously.

Additionally, in order to make the geometric consistency stage invariant to flip transformation, we have reintroduced a unified method that incorporates the original and flipped versions.

From the results, different from object recognition, increasing the distinctive power of descriptor by extending the length of the feature vector does not affect the accuracy of detection positively. The main reason is that since we utilize a quantization-based indexing procedure on feature vector, the number of bit per component of vector would be decreased for longer representation. Hence, the amendment with preserving the length of the vector like flip invariant version is more suitable on this task.

Also spatial descriptors outperform spatio-temporal descriptors. The main reason is that unlike action recognition, the static trajectories would give distinctive information. However, exploiting motion content becomes useless for this assumption.

6.2 Future Work

In the future, we will continue to investigate spatio-temporal features on this task. We believe that combining spatial content with time variations can have more distinctive information than pure spatial signature.

PUBLICATIONS

* S. Özkan, E. Esen and G.B. Akar, "Performance analysis of local indexing methods for video copy detection.", Signal Processing and Communication Application Conference (SIU), 2014. (In Turkish)

* S. Özkan, E. Esen and G.B. Akar, "Visual group binary signature for video copy detection.", International Conference on Pattern Recognition (ICPR), 2014.

* S. Özkan, E. Esen and G.B. Akar, "Enhanced spatio-temporal video copy detection by combining trajectory and spatial consistency.", International Conference on Image Processing (ICIP), 2014.

REFERENCES

[1] A. Saraçoğlu, E. Esen, T.K. Ateş, B.O. Acar, U. Zubari, E. C. Ozan, E. Özalp, A.A. Alatan and T. Çiloğlu, "Content based copy detection with coarse audio-visual fingerprints.", International Workshop on Content-Based Multimedia Indexing, pp. 213-218, 2009.

[2] C. Kim and B. Vasudev, "Spatiotemporal sequence matching for efficient video copy detection.", IEEE Transaction on CSVT, pp. 127-132, 2000

[3] M.C. Yeh and K.T. Cheng, "Video copy detection by fast sequence matching.", Proceeding of ACM International Conference on Image and Video Retrieval, 2009

[4] M.C. Yeh and K.T. Cheng, "A compact, effective descriptor for video copy detection.", Proceeding of ACM International Conference on Multimedia, 2009.

[5] K. Taşdemir and E. Çetin, "Motion vector based features for content based video copy detection.", International Conference on Pattern Recognition, 2010.

[6] R. Roopalakshmi and G.R.M. Reddy, "A novel CBCD approach using MPEG-7 motion activity descriptors.", International Symposium on Multimedia, 2011.

[7] O. Küçüktunç, M. Baştan, U. Güdükbay and Ö. Ulusoy, "Video copy detection using multiple visual cues and MPEG-7 descriptors.", Journal of Visual Communication and Image Representation, vol. 21, iss. 8, 2010.

[8] T.K. Ateş, A. Saraçoğlu, M. Soysal, Y. Turgut, O. Oktay and A.A. Alatan, "Content based video copy detection with local descriptors.", Signal Processing and Communications Applications Conference, pp. 49-52, 2010.

[9] E. Maani , S. A. Tsaftaris and A.K. Katsaggelos, "Local feature extraction for video copy detection in a database.", International Conference on Image Processingi pp. 1716-1719, 2008.

[10] M. Heritier, V. Gupta, L. Gagnon, G. Boulianne, S. Foucher and P. Cardinal, "Crim's content based copy detection system for TRECVID.", TRECVID 2009 Workshop, 2009.

[11] J. Law-To, O. Buisson, V. G. Brunet and N. Boujemaa, “Robust voting algorithm based on labels of behavior for video copy detection.”, Proceeding of ACM International Conference on Multimedia, pp. 835-844, 2006.

[12] X. Wu, Y. Zhang, Y. Wu, J. Guo and J.Li, “Invariant visual patterns for video copy detection.”, International Conference on Pattern Recognition, 2008.

[13] G. Willems, T. Tuytelaars and L.V. Gool, “Spatio-temporal features for robust content-based video copy detection.”, Proceeding of ACM International Conference on Multimedia, 2008.

[14] Z. Liu, T. Liu and B. Shahraray, “AT&T research at TRECVID 2009 content-based copy detection.”, TRECVID 2009 Workshop, 2009.

[15] E. Younessian, X. Anguera, T. Adamek, N. Oliver and D. Marimon, “Telefonica research at TRECVID 2010 content-based copy detection.”, TRECVID, 2010.

[16] Y. Uchida, M. Agrawal and S. Sakazawa, “Accurate content-based video copy detection with efficient feature indexing.”, Internation Conference on Multimedia Retrieval, 2011.

[17] M. Douze, H. Jegou and C. Schmid, “An image-based approach to video copy detection with spatio-temporal post filtering.”, IEEE Transaction on Multimedia, pp. 257-266, 2010.

[18] J. Law-To, L. Chen, A. Joly, I. Laptev, O. Buisson, V.G. Brunet, N. Boujemaa and F. Stentiford, “Video copy detection: a comparative study.”, Proceeding of ACM International Conference on Image and Video Retrieval, pp 371-378, 2007.

[19] G.C. Langelaar, I. Steyawan and R.L. Lagendijk, “Watermarking digital image and video data. A state-of-the-art overview.”, IEEE Signal Processing Magazine, pp.20-46, 2000.

[20] S.F. Chang, W. Chen, H.J. Meng, H. Sundaram and D. Zhong, “VideoQ: an automated content-based video search system using visual cues.”, Proceeding of ACM International Conference on Multimedia, pp. 313-324, 1997.

[21] J. Graham, D.G. Stork, “Content-based web advertising.”, US Patent No. 6,804,659, 2004.

[22] D.Q. Zhang and S.F. Chang, “Detecting image near-duplicate by stochastic attributed relational graph matching with learning.”, Proceeding of ACM International Conference on Multimedia, pp. 877-884, 2004.

[23] K. Mikolajczyk and C. Schmid, “A performance evaluation of local descriptors.”, IEEE Transaction on Pattern Analysis and Machine Intelligence, pp. 1615-1630, 2005.

[24] T. Lindeberg and J. Garding, “Shape-adapted smoothing in estimation of 3D shape cues from affine deformations of local 2D brightness structure.”, Image and Vision Computing, pp. 415-434, 1997.

[25] D.G. Lowe, “Distinctive image features from scale-invariant keypoints.”, International Journal of Computer Vision, pp.91-110, 2004.

[26] K.E.A. van de Sande, T. Gevers and C.G.M. Snoek, “Evaluating color descriptors for object and scene recognition.”, IEEE Transaction on Pattern Analysis and Machine Intelligence, pp. 1582-1596, 2010.

[27] W.L. Zhao and C.W. Ngo, “Flip-invariant SIFT for Copy and Object Detection.”, IEEE Transaction on Image Processing, pp. 980-991, 2013.

[28] H. Bay, T. Tuytelaars and L.V. Gool, “Surf: Speeded up robust features.”,European Conference on Computer Vision, pp. 404-417, 2006.

[29] H. Wang, A. Klaser, C. Schmid and C.L. Liu, “Dense trajectories and motion boundary descriptors for action recognition.”, Internation Journal of Computer Vision, pp. 60-79, 2013.

[30] J. Shi and C. Tomasi, “Good features to track.”, Internation Conference on Computer Vision and Pattern Recognition, pp. 593-600, 1994.

[31] G. Farneback, “Two-frame motion estimation based on polynomial expansion.”, Image Analysis, pp. 363-370, 2003.

[32] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection.”, International Conference on Computer Vision and Pattern Recognition, pp. 886-893, 2005.

[33] N. Dalal, B. Triggs and C. Schmid, “Human detection using oriented histograms of flow and appearance.”, European Conference on Computer Vision, pp. 428-441, 2006.

[34] R. Yayesand and B. Neto, “Model information retrieval.”, ACM Press, 1999.

[35] I. Laptev, “On space-time interest points.”, International Journal of Computer Vision, pp. 107-123, 2005.

[36] I. Laptev, M. Marszalek, C.Schmid and B. Rozenfeld, “Learning realistic human actions from movie.” , International Conference on Computer Vision and Pattern Recognition, pp. 1-8, 2008.

[37] M.A. Turk and A.P. Pentland, “Face recognition using eigenfaces.”, International Conference on Computer Vision and Pattern Recognition, pp. 586-591, 1991.

[38] V. Deepu, S. Madhvanath, A.G. Ramakrishnan, “Principle component analysis for online handwritten character recognition.”, International Conference on Pattern Recognition, pp 327-330, 2004.

[39] Group TMPE (2001), Mpeg-7 multimedia content description interface ISO/IEC 15938.

[40] M. Soysal, K.B. Loğoğlu, M. Tekin, E. Esen, B.O. Acar, E.Z. Ozan, T.K. Ateş, H. Sevimli, M. Sevinç, İ. Atıl, S. Özkan, M.A. Arabacı, S. Tankız, T. Karadeniz, D. Önür, S. Selçuk, A.A. Alatan and T. Çiloğlu, “Multimodal concept detection in broadcast media: KavTan.” Journal of Multimedia Tools and Applications, pp. 1-46, 2013.

[41] S. Lazenbnik, C. Schmid and J. Ponce, “Beyond bags of features: Spatial pyramid matching for recogniting natural scene categories.”, International Conference on Computer Vision and Pattern Recognition, pp. 2169-2178, 2006.

[42] R. Arandjelovic and A. Zisserman, “Three things everyone should know to improve object retrieval.”, International Conference on Computer Vision and Pattern Recognition, pp. 2911-2918, 2012.

[43] J.R.R. Uijlings, A.W.M. Smeulders and R.J.H. Scha, “Real-time visual concept classification.”, IEEE Transaction on Multimedia, pp. 665-681, 2010.

[44] E. Tola, V. Lepetit and P. Fua, “Daisy: An efficient dense descriptor applied to wide-baseline stereo.”, IEEE Transaction on Pattern Analysis and Machine Intelligence, pp.815-830, 2010.

[45] J. Sivic and Zisserman, “Video google: a text retrieval approach to object matching in video.”, International Conference on Computer Vision, pp 1470-1477, 2003.

[46] J. Yang, K .Yu, Y. Gong and T. Huang, “Linear spatial pyramid matching using sparse coding for image classification.”, pp. 1794-1801, 2009.

[47] J. Wang, J. Yang, F. Lv, T. Huang and Y. Gong, “Local-constrained linear coding for image classification.”, International Conference on Computer Vision and Pattern Recognition, pp. 3360-3367, 2010

[48] H. Jegou, M. Douze and C. Schmid, “Hamming embedding and weak geometric consistency for large scale image search.”, European Conference on Computer Vision, pp. 304-317, 2008.

[49] H. Jegou, M. Douze and C. Schmid, “Product quantization for nearest neighbor search.”, IEEE Transaction on Pattern Analysis and Machine Intelligence, pp. 117-128, 2011.

[50] M.A. Fischler and R.C. Bolles, “Random sample consensus.”, Communition of ACM, pp. 381,395, 1981.

[51] J. Philbin, O. Chum, M. Isard, J. Sivic and A. Zisserman, “Object retrieval with large vocabularies and fast spatial matching.”, International Conference on Computer Vision and Pattern Recognition, pp. 1-8, 2007.

[52] R.O. Duda, P.E. Hart, D.G. Stork, “Pattern Classification.”, John Wiley and Sons, 2012.

[53] D. Arthur, s. Vassilvitskii, “kmeans++: the advantages of careful seeding.”, Proceeding of ACM-SIAM Symposium on Discrete Algorithms, pp. 1027-1035, 2007.

[54] W.L. Zhao, X. Wu and C.W. Ngo, “On the annotation of web videos by efficient near-duplicate search.”, IEEE Transaction on Multimedia, pp. 448-461, 2010.

[55] B. Özdemir and S. Aksoy, “Image classification using subgraph histogram representation.”, International Conference on Pattern Recognition, 2010.

[56] Y. Zhang, Z. Jia and T. Chen, “Image retrieval with geometry-preserving visual phrases.”, International Conference on Computer Vision and Pattern Recognition, pp. 809-816, 2011.

[57] L. Dai, X. Sun, F. Wu and N. Yu, “Large scale image retrieval with visual groups.”, International Conference on Image Processing, 2013.

[58] A.F. Smeaton, P. Over and W. Kraaih, “Evaluation campaigns and TRECVid.”, International Workshop on Multimedia, 2006.