

Master Thesis

# Video fingerprinting in compressed domain

Fingerprinting method based on motion vector

## Thesis Committee:

Prof. dr. ir. R.L. Lagendijk

Dr. A. Koz

Dr. ir. R. Heusdens

Dr. C.P. Botha

Author	<b>Muhammad AlBaqir</b>
Email	arabtm@yahoo.com
Student number	1330543
Thesis supervisor	Prof. dr. ir. R.L. Lagendijk Dr. A. Koz
Date	19 June 2009

*To my beloved family*

*Maryunis, Abdillah, Ali and Navisah*

## Table of contents

<b>List of figures .....</b>	<b>iv</b>
<b>List of tables .....</b>	<b>v</b>
<b>Preface .....</b>	<b>vi</b>
<b>Acknowledgment .....</b>	<b>vii</b>
<b>Chapter 1 – Introduction .....</b>	<b>1</b>
<b>Chapter 2 – Existing methods and research contribution .....</b>	<b>4</b>
2.1. Review of existing video fingerprinting methods .....	4
2.1.1. Pre-processing .....	4
2.1.2. Framing .....	5
2.1.3. Feature extraction .....	6
2.1.4. Post-processing .....	7
2.1.5. Similarity measurement .....	7
2.2. Research contribution .....	7
<b>Chapter 3 - Video fingerprinting in compressed domain .....</b>	<b>10</b>
3.1. MPEG-1 motion vectors .....	11
3.2. Motion field generation from MPEG-1 motion vectors .....	12
3.3. Video fingerprinting based on motion field .....	15
3.4. Similarity measurement .....	17
<b>Chapter 4 Experimental Setup .....</b>	<b>19</b>
4.1. Data set descriptions .....	19
4.2. System design parameters .....	20
4.3. Temporal misalignment experiment .....	21
4.4. Video distortion experiment .....	22
4.5. Multiple fingerprints query experiment .....	24
4.6. Performance measurements .....	25
<b>Chapter 5 – Experimental Results .....</b>	<b>28</b>
5.1. Experiments on design parameters .....	28
5.1.1. Zero-motion MB exclusion .....	29
5.1.2. Experiment on frame-grouping size .....	30
5.1.3. Experiment on histogram bin size .....	32

5.2. Experiment on temporal misalignment .....	33
5.3. Experiments on different types of video distortion.....	35
5.3.1. <i>Different bitrate</i> .....	35
5.3.2. <i>Different GOP structure</i> .....	37
5.3.3. <i>Different GOP structure and bitrate</i> .....	38
5.3.4. <i>Contrast adjustment</i> .....	39
5.3.5. <i>Brightness adjustment</i> .....	40
5.3.6. <i>Subtitling</i> .....	40
5.3.7. <i>Video scaling</i> .....	40
5.3.8. <i>Video cropping</i> .....	41
5.3.9. <i>Video frame spatial shifting</i> .....	42
5.4. Experiment on multiple fingerprints query .....	43
<b>Chapter 6 – Discussions and open issues.....</b>	<b>44</b>
6.1. Discussions .....	44
6.2. Open issues .....	45
<b><i>Bibliography</i> .....</b>	<b>47</b>

## *List of figures*

<i>Figure 1.1: Video fingerprinting system general framework.</i>	2
<i>Figure 2.1: Video fingerprinting main stages.</i>	4
<i>Figure 2.2: spatial framing</i>	6
<i>Figure 2.3: luminance intensity centroid (<math>X_c</math>, <math>Y_c</math>) generation, as proposed in [5].</i>	6
<i>Figure 2.4: DC coefficient interpolation, as proposed in [11].</i>	9
<i>Figure 3.1: General framework of video fingerprint generation.</i>	10
<i>Figure 3.2: temporal-referencing in MPEG-1 compression.</i>	12
<i>Figure 3.3: Two consecutive group of picture (<math>M = 10</math>, <math>N = 2</math>).</i>	13
<i>Figure 3.4: MPEG-1 motion vectors in video frame temporal order.</i>	13
<i>Figure 3.5: Motion direction histogram generation procedure, from a), b) to c).</i>	16
<i>Figure 3.6: Frame-grouping with the size of <math>L=5</math>, performed for <math>F = 15</math> video frames, resulted in <math>K = 3</math> fingerprints <math>H_1</math>, <math>H_2</math> and <math>H_3</math> respectively.</i>	17
<i>Figure 3.7: Histogram intersection method. Performing histogram intersection for a) and b) result in histogram c).</i>	18
<i>Figure 4.1: Temporal misalignment between database and query video.</i>	21
<i>Figure 4.2: Difference between single a) and multiple fingerprints query b) matching process, where <math>R</math> and <math>Q</math> are the fingerprints for database video and query video, respectively.</i>	25
<i>Figure 4.3: intra-statistics calculation.</i>	26
<i>Figure 4.4: inter-statistics calculation.</i>	26
<i>Figure 4.5: Probability distribution of PD and PS</i>	27
<i>Figure 4.6: ROC curve.</i>	27
<i>Figure 5.1: Two perceptually different video of 300 frames</i>	29
<i>Figure 5.2: Histogram intersection values between two videos given frame-by-frame comparison with zero-motion MB a) or not b)</i>	29
<i>Figure 5.3: ROC curve for different frame-grouping size</i>	30
<i>Figure 5.4: The comparison of two different videos' histogram intersection values distribution a) PD and b) PS given different frame-grouping size.</i>	31
<i>Figure 5.5: TPR values for different bin size, given different frame-grouping size</i>	32
<i>Figure 5.6: Motion direction histograms, with different bin size of a) 8, b) 16 and c) 32</i>	32

<i>Figure 5.7: Motion direction quadrant according to the search range with a) divided into 4 quadrants and b) divided into 8 quadrants.....</i>	<i>33</i>
<i>Figure 5.8: Temporal misalignment experiment procedure.....</i>	<i>34</i>
<i>Figure 5.9: Temporal shift from -12 to +12 frame shifts, with frame-grouping size 25.....</i>	<i>34</i>
<i>Figure 5.10: Average ROC curve of unidentified temporal shift, given two different frame-grouping sizes of 25 and 100 .....</i>	<i>35</i>
<i>Figure 5.11: Different bitrate experiment results .....</i>	<i>36</i>
<i>Figure 5.12: motion field distribution of a video sequence with different bitrates.....</i>	<i>36</i>
<i>Figure 5.13: Macroblock prediction type distribution of a video sequence with different bitrates.....</i>	<i>37</i>
<i>Figure 5.14: Different GOP structure experiment results.....</i>	<i>38</i>
<i>Figure 5.15: Different GOP structure and bitrate experiment results.....</i>	<i>38</i>
<i>Figure 5.16: Contrast adjustment experiment results .....</i>	<i>39</i>
<i>Figure 5.17: Brightness adjustment experiment results .....</i>	<i>40</i>
<i>Figure 5.18: Video scaling experiment results .....</i>	<i>41</i>
<i>Figure 5.19: Video cropping experiment results .....</i>	<i>41</i>
<i>Figure 5.20: Multiple fingerprints query with temporal frame shifting experiment results.....</i>	<i>43</i>

## *List of tables*

<i>Table 3.1: Motion calculation for each combination of past MB and current MB. ....</i>	<i>14</i>
<i>Table 4.1: Characteristics of the original videos. ....</i>	<i>19</i>
<i>Table 4.2: Programming tools utilized during the experiments.....</i>	<i>20</i>
<i>Table 4.3: Description for each types of video distortion that are applied.....</i>	<i>22</i>
<i>Table 4.4: Types of video distortion that are considered.....</i>	<i>23</i>

## *Preface*

This thesis report serves as the documentation of my final project as a master student at Information and Communication Theory (ICT) group, Delft University of Technology. This project commences from May 2008 until June 2009.

*Muhammad AlBaqir,  
Delft, 19 June 2009*

## Acknowledgment

I fully dedicate this thesis to my beloved mother for her commitment and consistency in shaping me to become what I am. Despite my stubbornness, she never gives up in motivating me and pointing the right direction in life. Obviously, I'm also grateful for other members of my family. I would like to thank my father for his moral and spiritual support, my older brother, whom I took examples from, even though he never realizes it and my younger sister for just being such a lovely sister.

For the completion of my master thesis, I would like to thank professor Lagendijk for his honest and insightful supervision. I genuinely consider him as my best academic teacher so far. His vast knowledge, particularly in multimedia signal processing, never ceases to impress me. I would also like to thank Alper for being my daily supervisor. He was always gracious during our informal discussions and I really appreciate his encouragements when I was lost in thesis wonderland. I have considered him more of as a friend rather than just a supervisor.

Ever since I arrived in Delft, I've met people whom I indebted for their friendship and assistance. I would like to acknowledge the following people whom have supported me during my study period as a master student at TU Delft. Habib & Linda for being wonderful friends ever since day one. Edy for being the most compassionate neighbor and allowing me to be his roommate for 6 months in his new place, after we got kicked-out from our old housing. Angga for being tremendously helpful and trustworthy friend in every occasion. Bian for sharing my craziness and being an exceptionally consistent friend. Jo for being my party friend and sharing the same hobby of playing billiard. Narto & Nani for being such a nice housemates and allowing me to dine with them particularly during the last hectic days of my thesis. Yusuf for giving me a lifetime warranty to repair my antique bicycle. Ikhsan for being the most helpful senior for discussing my work. Firman for allowing me to stay in his place for two months when I was homeless and also for lending me his laptop when mine was not functioning properly. Tisha for being such a nice lab companion during the last weeks of my project and also for lending me her laptop, which I'm currently typing on at the moment (*Oh man, I borrowed two laptops already. I really should fix my laptop!*). And last but not the least, Dedy for giving me frequent dinner invitation and also for allowing me to stay in his place for two weeks when I was temporally kicked-out out by Edy. Thank you so much guys!

-Baqir-



## Chapter 1 – Introduction

The simplicity of today's digital technology makes it easier for anyone to produce, edit or duplicate video content. With the help of digital media distribution such as the internet, consumers are faced with vast amount of video content that are generated by video producers. Consequently, a means of content management is tremendously required (*e.g. how to efficiently find a video*).

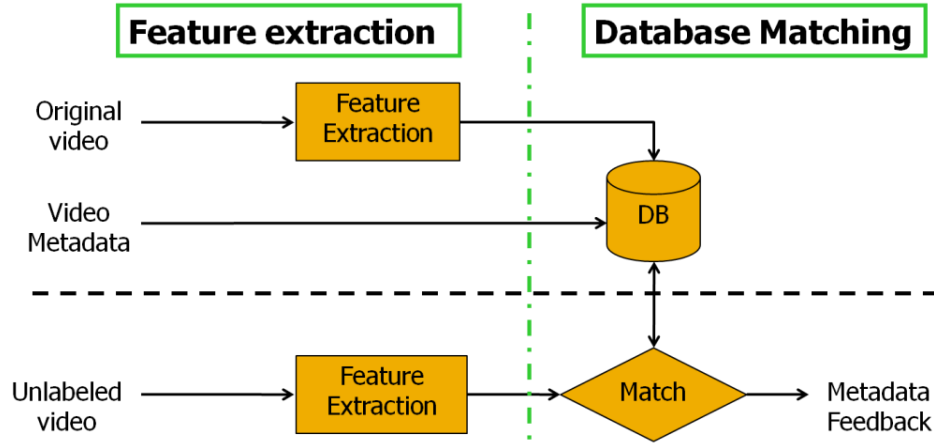
The most common procedure for managing video content is by analyzing the tagged information within the data (*i.e. metadata*). However, there is no guarantee that the metadata contains sufficient descriptive information of the content. This is due to the fact that metadata can be easily modified, either intentionally or not. Such metadata modification might happen due to video re-encoding or deliberately removed to avoid identification. Due to this notion, several approaches are proposed for uniquely identifying a video, assuming that the proper video metadata is insufficient.

The first approach is cryptographic hashing, where a video content is mapped into a unique hash value. In this method, identification process can be performed by examining the hash value. To take into account, this method is not applicable for identifying perceptually similar video. Given two videos of same content but with slight distortions (*e.g. cropping, subtitling, different bitrate*), the generated hash values are totally different. Therefore, these two videos are considered different even though they contain the same information.

The second approach is known as watermarking, where a unique label is embedded into a video content. In this approach, the embedded data is generally robust such that its removal would be difficult. Subsequently, the identification process can be performed by extracting the video watermark. Nevertheless, watermarking has to be performed prior to broadcasting the video to public. Therefore, video that has been released, which does not have watermark in it, is impossible to identify in such an approach.

As an alternative for the previous mentioned approaches, video fingerprinting method is proposed. Similar to cryptographic hashing, video is also represented with hash value derived from its content. However, this value is robust against perceptual changes applied to video content. Therefore, video fingerprinting is also known as perceptual hashing method. Furthermore, in comparison with watermarking, no data embedding is required since the content itself would be used to generate the video's discriminative feature.

For performing identification, video fingerprinting method should extract every original video's fingerprint and stored it inside the database, along with the corresponding metadata. Therefore, given an unknown query video, the system is able to perform similarity measurement between query video's extracted fingerprint and the original fingerprints. If a match is found, the relevant information regarding the query video is then presented. An illustration of video fingerprinting method is shown in *figure 1.1*.



*Figure 1.1: Video fingerprinting system general framework.*

In order to perform effectively, a video fingerprinting method should satisfy several parameters [1]. The main parameters are:

- **Robustness:** Any type of distorted video should be identified, given the video is perceptually similar with the original video.
- **Uniqueness:** Two different video should not be considered as perceptually similar. Therefore, the generated fingerprints should be distinctive enough to characterize the video content.
- **Granularity:** This parameter refers to the sufficient length of a video that is required to perform identification. It is expected that any video fingerprinting method should only require a small amount of time length (*i.e. several seconds*) to perform video identification.
- **Complexity:** The amount of processing needed to perform fingerprint extraction and comparison should not take a lot of computational power and time.

Video fingerprinting can be applied to different type of digital video management scenarios. Video sharing website such as YouTube does not allow their users to upload copyrighted materials. However, YouTube stated that the upload rate of their server is up to 20 hours of video materials per minute [2]. Therefore, a video fingerprinting system to automatically identify the uploaded video is really

needed. Another usage of video fingerprinting is in broadcast monitoring. The advertisers need to monitor their commercials automatically in the broadcasting stream, whether it is displayed according to the agreement or not. Video fingerprinting can also be applied for video identification in peer-to-peer (*P2P*) network, in which it has become one of the most popular methods for transferring video materials. Such applications in P2P network can be content labeling, content distribution statistics or content removal from the network.

Most circulating videos are in compressed form, in which the quantity of the video data is reduced to obtain sufficient bit size. The reason for performing video compression is due to video application's limitation in processing large video size (*e.g. limited bandwidth, small storage size, fast transmission requirement*). Due to the pervasiveness of compressed video, a video fingerprinting method should also be able to uniquely identify any compressed video.

Conventional video fingerprinting method extracts fingerprints by analyzing the video frame as an image. Therefore, given a compressed video, decompression is required to rearrange video frames from compressed domain into spatial domain. Due to decompression process, a certain amount of time is added to the entire process of fingerprinting generation. Given this notion, this thesis aims to investigate the possibility of extracting video fingerprints directly from the compressed video, without performing full decompression.

The structure of this thesis is as follows. Chapter 2 discusses on several known fingerprinting method and our research contribution to generate video fingerprint from compressed video. Chapter 3 presents the full framework of our proposed method. In addition, the general concept of video compression is also discussed in this chapter. Chapter 4 explains the evaluation tests for our proposed method. Chapter 5 presents the results and analysis of all evaluation tests. Finally, discussions regarding the performance of the proposed method are given in chapter 6. Furthermore, open issues regarding our approach are also discussed in this chapter.

## Chapter 2 – Existing methods and research contribution

In this chapter, further descriptions are given regarding video fingerprint method in order to provide better comprehension. Furthermore, our proposed video fingerprinting method is also presented within this chapter. In the first section, a brief discussion is given regarding existing video fingerprinting processes. Subsequently, this chapter is further divided into subsections, in which each subsections give more details regarding the general stages of video fingerprinting method. In the second section, we present the motivation for proposing our video fingerprinting method that utilize feature extracted from compressed video.

### 2.1. Review of existing video fingerprinting methods

In general, video fingerprinting method can be divided into two fundamentals processes, which are fingerprint generation and similarity measurement. Furthermore, fingerprint generation also consist of several steps, which are *pre-processing*, *framing*, *feature extraction* and *post-processing*. As an illustration, consider the following figure 2.1. Each fingerprint generation steps and also similarity measurement are explained in the following subsections, along with examples from the video fingerprinting methods proposed in the literature.

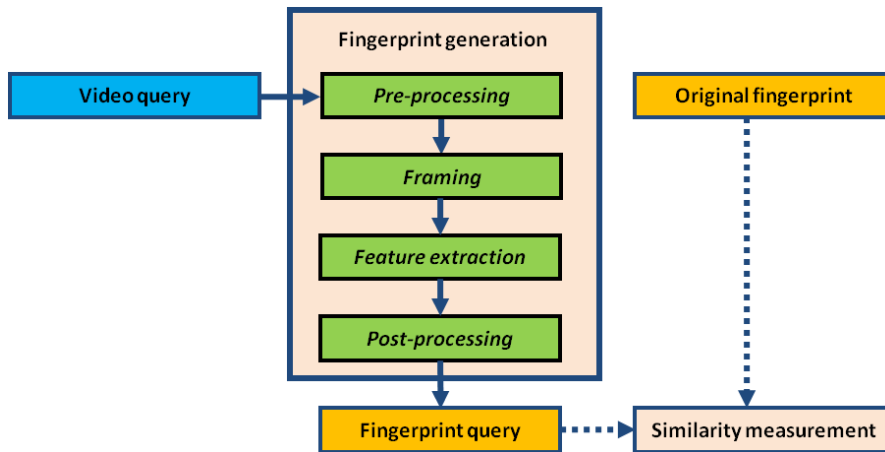


Figure 2.1: Video fingerprinting main stages.

#### 2.1.1. Pre-processing

Given a query video, video fingerprinting method would first perform pre-processing step, in which the video is converted into a general format. This procedure is carried out to achieve robustness

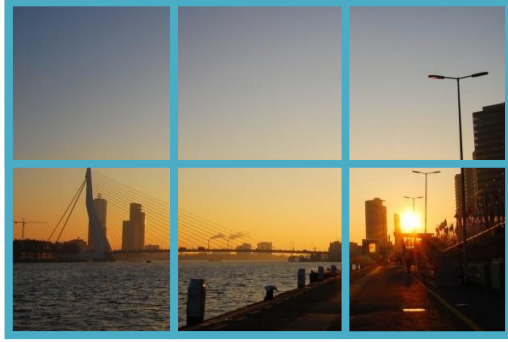
against different type of video formats and characteristics. The following are several examples of pre-processing steps mentioned in the literature:

- *Analog to digital conversion*: Since video can also be stored in an analog format, digitalization is required in order to extract the video fingerprint
- *Frame gray scaling*: In order to adapt to color variations between similar videos, the color of every frames are converted into grayscale [3] [4].
- *Frame size normalization*: Since video can have different aspect ratio, several fingerprinting methods perform frame size normalization into a preset size, in order to extract the relevant feature [3] [5-7]. This procedure is common in fingerprinting method that extracts the feature from spatial blocks, as explained in the following subsection.
- *Video decompression*: Compressed video needs to be decompressed in order to extract fingerprint feature from the video frame.

### 2.1.2. Framing

Framing is performed for video content, in which the video is divided into smaller segment. This segmentation can be both in spatial domain or temporal domain. In spatial domain, framing is commonly performed by dividing the video frames into spatial blocks [3] [5-7]. By performing spatial blocking, a more compact video frame representation is achieved from each block, since every pixel within the block is averaged (*i.e. reducing spatial redundancy*). In addition, spatial framing can also be performed by using radial projection lines, as proposed in [8] [9]. This type of spatial framing is considered in order to achieve robustness against geometrical distortion (*e.g. scaling, rotation*). From these spatial blocks, the fingerprint feature is extracted. Illustrations of spatial framing are presented in the following *figure 2.2*.

Framing in temporal domain can also be performed to reduce temporal redundancy. Due to the consideration that consecutive video frames can still have spatial similarity. Several methods proposed to represent similar consecutive frames as a single key-frame [3] [5] [9]. Afterwards, the fingerprint feature is then extracted from the generated key-frames.



a) Rectangular spatial blocking



b) Radial projection lines [8] [9]

Figure 2.2: spatial framing.

### 2.1.3. Feature extraction

The purpose of feature extraction is to obtain fingerprint feature that is robust against different type of video distortions. A video fingerprint feature can be obtained from spatial or spatial-temporal dimension. In a spatial feature extraction, the generated feature is obtained by only considering a single frame (*i.e.* or *key frame*). Moreover, spatial-temporal feature extraction also considers the spatial feature difference between consecutive frames.

In general, luminance value is considered sufficiently representative as fingerprint feature, in which different method applied different approach to model the luminance value. In [3], [5] and [10] centroid of luminance intensity is calculated for every spatial blocks. An illustration is given in *figure 2.3*. *Massoudi et.al* [9] also considers luminance intensity as well as the direction of the radial projection lines. *Oostveen et.al* [7] proposed to calculate the mean luminance difference between two consecutive frames. *Hamon et.al* [4] proposed to quantify the luminance intensity of every key frame in a histogram.



Figure 2.3: luminance intensity centroid ( $X_c$ ,  $Y_c$ ) generation, as proposed in [5].

#### 2.1.4. Post-processing

Given the extracted fingerprint feature, post-processing is applied in order to obtain the video fingerprint. This step is also used to further reduce the extracted feature's redundancy. The common procedure is to model the generated feature into a fixed set of values. Oostveen *et.al* [7] proposed to quantify the extracted feature by analyzing the value's sign (*i.e. positive or negative*). Given a positive sign, the value is quantized as 1 and 0 given a negative sign. Hamon *et.al* [3] proposed to quantify the histogram value in a binary form, using the median gray value as the threshold. Li *et.al* [5] maps the centroid value of each spatial block into a hash vector, using predefined hashing function. Similar mapping approaches are also proposed in [3], [9] and [10].

#### 2.1.5. Similarity measurement

To determine whether two fingerprints are perceptually similar or not, a similarity measurement is required. In general, the generated fingerprints are a sequence of a fixed set of values. Therefore, Hamming distance is generally used to obtain the number of positions in which the hash values are different [4] [7]. If the fingerprint sequence is in binary form, the Hamming distance is often known as bit error rate (*BER*). Other measurement that is used for identifying fingerprint similarity is distance metric, such as Euclidean distance used in [9] and [10].

Similarity measurement result is analyzed by evaluating the false positive rate and true positive rate of the system implementation. False positive rate determines the probability in which two different videos are detected as perceptually similar. On the other hand, True positive rate determines the probability in which the system is able to correctly identify a perceptually similar video. To sum up, a good video fingerprinting method should have a very low false positive rate while still maintaining a high true positive rate.

### 2.2. Research contribution

In general, videos contain certain amount of data redundancy, either in spatial or temporal domain. For instances, a video redundancy can be image resemblance between consecutive frames or pixels similarity within a single frame. By decreasing data redundancy, a more efficient data rate can be achieved (*e.g. faster transmission rate, smaller video size*). In order to achieve this, most video applications perform compression towards the videos.

Video compression generally reduces data redundancy in spatial and temporal domain. Given spatial redundancy, the common procedure is to convert the video frames from the spatial domain to the frequency domain using discrete cosine transform (*DCT*). *DCT* is a well known process, in which most of image information is concentrated in the lower spatial frequencies, especially the zero frequency known as *DC coefficient*. Therefore, some of the less important information stored in *AC coefficients* can be discarded without having significant visible distortion.

In regards to temporal redundancy, the common compression procedure is to encode frame differences. In this process, a video frame is estimated from another frame, using motion estimation algorithm. Consequently, motion vectors that represent spatial shift between the blocks inside the neighboring frames are generated. Therefore, by only encoding motion vectors and the reference frame, more efficient data rate can be obtained rather than encoding each frame separately.

Several video fingerprinting methods have been proposed for identifying video, in which fingerprints are extracted by analyzing video in both spatial and temporal dimension. However, these conventional methods have one resemblance, in which video decompression is still required for extracting the fingerprint from a compressed video. In practical, faster computational time can be achieved if fingerprint is extracted directly from the compressed domain.

So far, only one method is known to propose video fingerprinting in compressed domain, in which *DC coefficient* is used to model the fingerprint [11]. Given the extracted *DC coefficient*, the video fingerprint method constructs the video frame, even though with a smaller video resolution. If *DC coefficient* is not present within a frame, an interpolation would be performed by using motion vectors and the *DC coefficients* from the reference frame, as illustrated in *figure 2.4*. Eventually, the modeled video frame is evaluated to obtain the key-frames of the video, which would be further analyzed for generating the fingerprints. Nevertheless, the performance result regarding this system is not presented within the literature. Therefore, no clear evaluation could be given regarding this method.



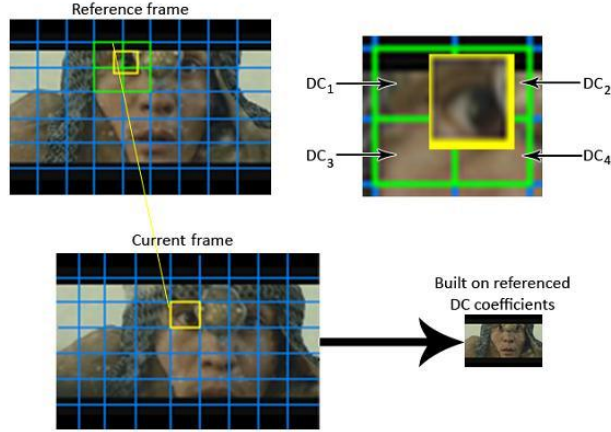


Figure 2.4: DC coefficient interpolation, as proposed in [11].

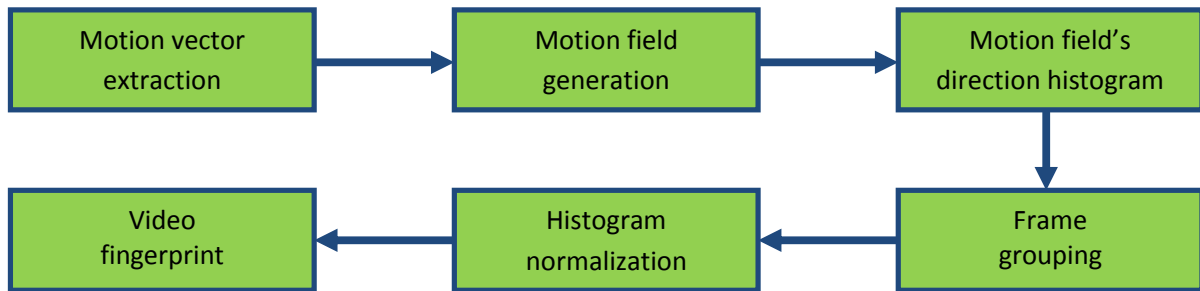
Alternatively, we proposed a video fingerprinting method, in which motion vectors are used to model the fingerprint. We consider utilizing motion vector since it is commonly generated during video compression for exploiting the temporal redundancy within a video. In this research, we analyze our method's performance using MPEG-1 video compression standard. Nevertheless, it is assumed that our method can also be applied to other compression type, given a certain modification. As a remark, our generated fingerprint cannot be considered as *full* fingerprint yet, since we do not go further in quantizing the generated fingerprint in a binary form.

To take into account, we realize that our method is unable to analyze compressed video that does not have motion vectors at all (*e.g. video consisting of still scenes, videos that are compressed by only considering spatial redundancy*). However, due to the assumption that most compressed video relies on temporal redundancy, analyzing motion vectors is still considered as an advantageous option.

## Chapter 3 - Video fingerprinting in compressed domain

This chapter describes our proposed video fingerprinting method that only requires partial-decompression given a compressed video. To generate the fingerprint, several steps need to be performed. To begin with, a relevant feature needs to be extracted from the compressed stream. We propose to exploit the motion vectors, which is one of the main parameters in video compression. These motion vectors are utilized to generate the motion fields, which are the motion between adjacent video frames.

In order to obtain a compact fingerprint size, we quantize the motion fields based on their motion direction, in a form of a histogram. Moreover, we also consider grouping several consecutive motion direction histograms as a single histogram, to further reduce the fingerprint size. Every generated histogram is then normalized by the total amount of motion fields that are utilized. This is required to attain the same scale between histograms. In the end, the normalized histogram is considered as the fingerprint of the compressed video. To illustrate the general framework of our proposed method, consider the following *figure 3.1*.



*Figure 3.1: General framework of video fingerprint generation.*

The organization of this chapter is as follows. Section 3.1 briefly discusses MPEG-1 video compression, which focuses more on motion vector generation. Section 3.2 explains how to transform the motion vectors that are extracted from MPEG-1 compressed video, into motion fields, which are the motions between consecutive video frames. Section 3.3 present our algorithm in modeling the motion fields as video fingerprint. Finally, Section 3.4 describes the similarity measurement, which is required to compare two fingerprints.

### 3.1. MPEG-1 motion vectors

In performing compression, MPEG-1 encoder normally classify the video frames into three types, namely I (*intra*) frame, P (*predicted*) frame and B (*bi-directional*) frame. Subsequently, each frame type is divided into macroblocks (*MB*) of 16x16 pixels. The coding scheme is then performed in macroblock basis. If intra coding is selected, the corresponding MB is encoded individually by exploiting the discrete cosine transform (DCT) coefficients. This is performed to reduce video frame's spatial redundancy. On the other hand, if inter coding is selected, the MB is then encoded using temporal reference (*i.e. referencing neighboring frames*), using motion estimation-motion compensation (*ME/MC*) algorithm. The purpose of using ME/MC is to reduce redundancy in temporal direction, since neighboring video frames could still be spatially similar.

An example on how *temporal-referencing* is performed is shown in *figure 3.2*. The general notion for *temporal-referencing* is that for a given MB, the encoder performs a search function within the reference frame (*i.e. either neighboring I-frame or P-frame, but not B-frame*) until it locates the most similar 16x16 pixels. The type of *temporal-referencing* is also different, depending on the types of MB, which are:

- *Intra MB*: Intra coded MB without any temporal reference. This MB can be found in every frame.
- *Forward MB*: Inter coded MB using temporal reference from *past* frame (*in display-order*). The motion vector in this coding is referred as *forward motion vector*. This MB can be found in P or B-frame.
- *Backward MB*: Inter coded MB using temporal reference from *future* frame. Correspondingly, the motion vector used in this coding is referred as *backward motion vector*. This MB can be found only in B frame.
- *Bi-directional MB*: Inter coded MB that uses both future and past frame as temporal references. Therefore, this type of MB has two motion vectors, which are *forward* and *backward motion vector*. This MB can be found only in B frame.

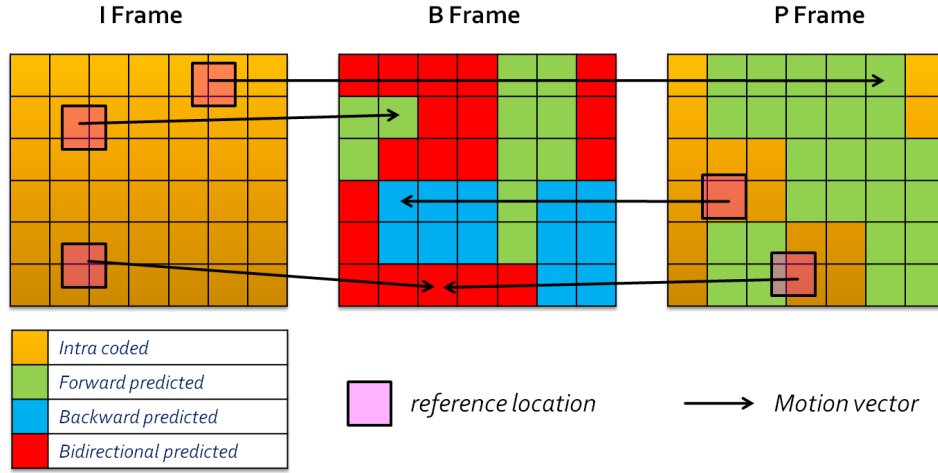


Figure 3.2: temporal-referencing in MPEG-1 compression.

After finding a match, a motion vector is then generated. Motion vector is represented with the value of  $\langle X, Y \rangle$  in which  $X$  constitute horizontal translation and  $Y$  as vertical translation. This motion vector represents the translation distance between the current frame MB location and the 16x16 location within the reference frame. Furthermore, using the motion vectors from previous frame, a *prediction motion vector* (PMV) is then computed. Along with the prediction error (*i.e. the difference between original MB and the estimated MB*), the difference between MV and PMV is then encoded. Therefore, motion vectors also characterize the video content, particularly in temporal direction.

In addition, there is also another type of macroblock, known as *skipped MB*. This MB can be found in P or B-frames, in which the characteristics are different depending on the frame type. In P-frame, a skipped MB is assumed as inter coded with zero motion vector (*i.e. can be considered as Forward MB with  $\langle 0, 0 \rangle$  motion vector*). Whereas in B-frames, it is assumed as inter coded and has the same motion vector as the previously encoded MB, which can be *forward, backward or bi-directional*.

### 3.2. Motion field generation from MPEG-1 motion vectors

In decoding process, motion vectors can be straightforwardly acquired from MPEG-1 compressed video stream, without too much processing (*i.e. partial decompression*). However, motion vectors do not really represent video content in a typical temporal sequence. This is due to MPEG-1 video frame structure pattern, generally known as *group of picture* (GOP), which is not fixed (*i.e. encoder dependent*). GOP structure is used in MPEG-1 compression in order to support both inter and intra-frame encoding.

GOP structures in general have one I-frame as the main reference for the other frames' macroblocks (*i.e. P and B-frames*) in the same structure. A GOP is usually represented with two figures,

$M$  and  $N$ , where  $M$  stands for the total frame, and  $N$  represents the number of B-frames between each two reference frames. An example of a GOP structure is shown in figure 3.3.

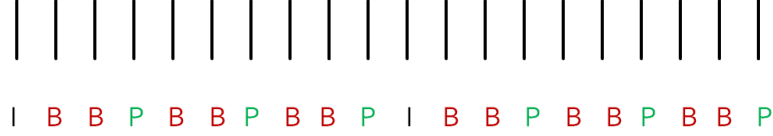


Figure 3.3: Two consecutive group of picture ( $M = 10$ ,  $N = 2$ ).

Motion vectors of a macroblock can only be generated by referencing either I or P-frames. Therefore, motion vectors are not always representing the motion between two consecutive frames (*e.g. two consecutive B-frames in figure 3.3*). However, it is possible to generate motion field using motion vectors.

As proposed in [12], given two consecutive frames that have the same reference frame, a motion field can be generated from their motion vectors by using simple calculation. To generate motion field between consecutive frames, calculation is performed in macroblock level by comparing each consecutive macroblock (*i.e. of two consecutive frames*) of the same spatial location. To give an illustration of motion field calculation, let us analyze the following figure 3.4.

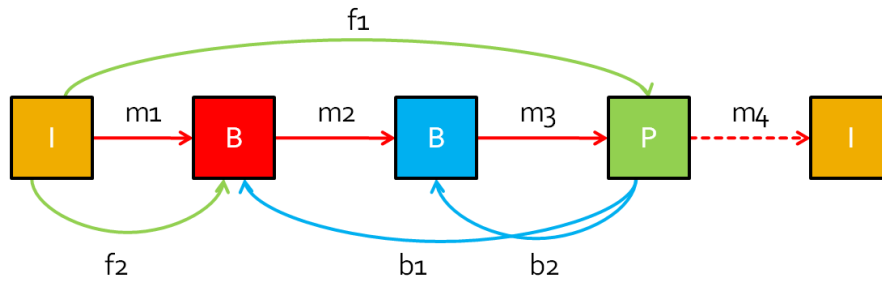


Figure 3.4: MPEG-1 motion vectors in video frame temporal order.

$m$ : motion fields,  $f$ : forward motion vectors,  $b$ : backward motion vectors,  $I, B, P$ : frame (macroblock) types.

Assuming that each video frame only contains the same type of macroblocks, the following calculations are performed for each motion field  $m$ :

1. Transition 1 ( $I \rightarrow B$ )

$m1 = f2$ , since  $f2$  represent the current MB's translation from the previous reference frame.

2. Transition 2 (**B** → **B**)

$m2 = -(b2 - b1)$ . Given that each current and past frame's MB have the same reference from the future P frame (i.e. *backward motion vectors b1 and b2 respectively*), the translation between these macroblocks is simply the difference between  $b1$  and  $b2$ . A negative sign is used, since the temporal direction is to the right.

3. Transition 3 (**B** → **P**)

$m3 = -b2$ . Similar with transition 1, backward motion vector  $b2$  is considered as the translation vector from *current* P-frame to *past* B-frame.

4. Transition 4 (**P** → **I**)

In this transition,  $m4$  could not be calculated, due to the motion vector that connects between these two frames is not present. I-frames do not have *temporal-referencing* while P-frames could not have *backward predicted* motion vectors from future reference frame.

To sum up, there can be 16 types of transition between macroblocks, since there are 4 types of MB in MPEG-1 compression. The following *table 3.1* gives the motion field calculation for each type of consecutive MB.

Past MB	Current MB	Motion calculation
Intra	Intra	<b>No Motion</b>
Intra	Forward	$(Fc - 0)$
Intra	Backward	<b>No Motion</b>
Intra	Bidirectional	$(Fc - 0)$
Forward	Intra	<b>No Motion</b>
Forward	Forward	$(Fc - Fp)$
Forward	Backward	<b>No Motion</b>
Forward	Bidirectional	$(Fc - Fp)$

Past MB	Current MB	Motion calculation
Backward	Intra	$(0 - Bp)$
Backward	Forward	<b>No Motion</b>
Backward	Backward	$(Bc - Bp)$
Backward	Bidirectional	$(Bc - Bp)$
Bidirectional	Intra	$(0 - Bp)$
Bidirectional	Forward	$(Fc - Fp)$
Bidirectional	Backward	$(Bc - Bp)$
Bidirectional	Bidirectional	$((Fc - Fp) + (Bc - Bp)) / 2$

Table 3.1: Motion calculation for each combination of past MB and current MB.

$Fc$  = forward current frame's motion vector,  $Fp$  = forward past frame's motion vector

$Bc$  = backward current frame's motion vector,  $Bp$  = backward past frame's motion vector

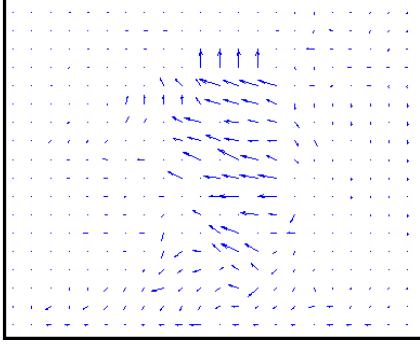
From *table 3.1*, it is shown that there are five possibilities where the method is not able to calculate the motion fields (*i.e.* “*No Motion*”). Although this result can be considered as a drawback, it is still possible to overcome it, as explained in the following section. To be taken into account, this method only focuses on motion vectors without regarding the prediction errors. To simplify the discussion, consecutive video frames are referred as “*video frames*” from this point forward.

Given the motion fields calculation presented in *table 3.1*, each MB could have three types of motion fields. Furthermore, the process of generating the fingerprint would be affected by this classification. The types of generated motion fields are:

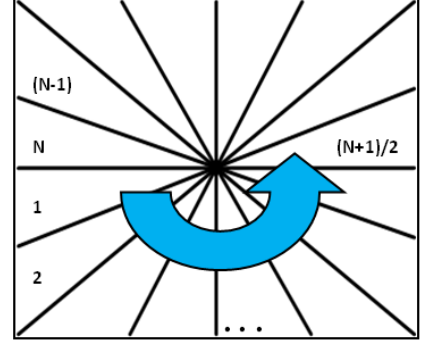
- *No-motion MB*  $\langle XX \rangle$ : This is the MB where no motion fields can be calculated, since the consecutive macroblocks are both intra coded or have different temporal reference frame.
- *Zero-motion MB*  $\langle 0, 0 \rangle$ : This is the MB where the motion vectors between past MB and present MB have the same value.
- *Moving MB*  $\langle X, Y \rangle$ : This is the MB where there are difference between motion vectors of past MB and present MB of  $\langle X, Y \rangle$ .

### 3.3. Video fingerprinting based on motion field

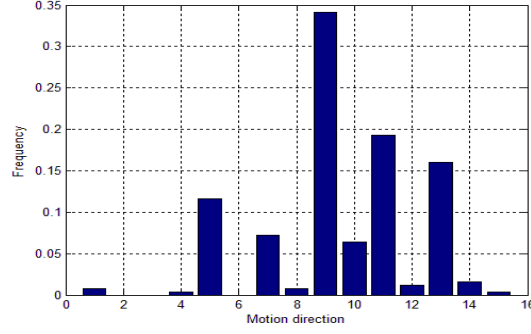
In order to analyze the generated motion fields, we propose to capture its motion direction and quantize it in a form of a histogram. To generate the histogram, motion field’s direction is quantized into  $N$ -quadrant, ranging from  $[-\Pi, \Pi]$ . For each quadrant, each histogram bin value corresponds to total macroblocks, whose motion field’s angle falls in the same quadrant. Essentially, the value of  $N$  represents the size of the video fingerprint, which corresponds to the total bin of the histogram. An illustration of this process is shown in *figure 3.5*.



a) Motion fields of an 18 x 22 MB video frame.



b) Calculate the number of motion field's direction occurrence per quadrant, ranging from  $[-\Pi, \Pi]$ .



c) Generated motion direction histogram per frame, given  $N = 16$

Figure 3.5: Motion direction histogram generation procedure, from a), b) to c).

From the three types of MB, given in the previous section, only *moving MB* is used in the histogram generation. The reason for this are:

- 1) *No-Motion MB*  $\langle XX \rangle$ : This type of MB does not contain any information at all regarding the motion of the video, since the motion field's calculation cannot be performed. Therefore, this MB does not contribute anything to the histogram generation.
- 2) *Zero-motion MB*  $\langle 0, 0 \rangle$ : Although this type of MB contains motion information of the video (*i.e. not moving can be considered as moving with zero translation*), it is not considered in the histogram generation. The reason is that our method only captures the motion direction, while *zero-motion MB*  $\langle 0, 0 \rangle$  does not contain any motion direction information whatsoever. Further explanation for this argument is presented in the fifth chapter of this report.



In our proposed method, histogram computation is applied not just for a single frame, but for a group of frames. This motivation for performing *frame-grouping* is to attain more compact fingerprint size. Suppose there are  $F$  frames, generating the histogram for every frame would result in total of  $K = F$  fingerprints. However, if frame grouping with the size of  $L$  is performed, lesser amount of fingerprint would be obtained, with the total of  $K = F/L$ . At the end, the generated histogram is normalized with the total amount of *moving MB*  $\langle X, Y \rangle$  within the scope of each frame-grouping. By performing normalization, every generated histogram has the same scale ranging from 0 to 1. Therefore, the similarity measurement can be performed straightforwardly, since the generated histogram always have the same scale.

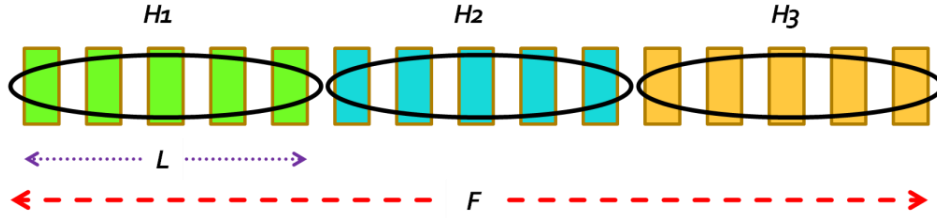


Figure 3.6: Frame-grouping with the size of  $L=5$ , performed for  $F = 15$  video frames, resulted in  $K = 3$  fingerprints  $H_1$ ,  $H_2$  and  $H_3$  respectively.

### 3.4. Similarity measurement

As explained in the previous section, the generated video fingerprint from our method is in the form of a normalized histogram, in which the summation of every histogram bin ranges from 0 to 1. Each histogram bin represents the total amount of MB that moves into a certain direction of a frame or grouped frames. In order to detect whether the video are perceptually similar or not, we consider using *histogram intersection* method [13] to compare the original fingerprint and the queried fingerprint.

Given two pair of fingerprints,  $Q$  and  $R$ , each containing  $N$  bins, the histogram intersection matching value  $W$  is defined as:

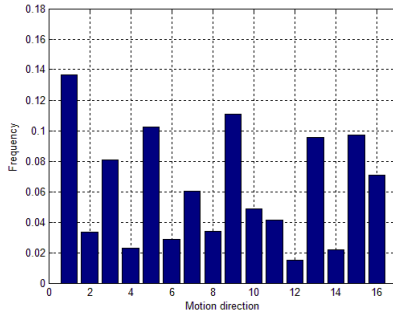
$$W(Q, R) = \sum_{i=1}^N \min(Q_{(i)}, R_{(i)})$$

In which,

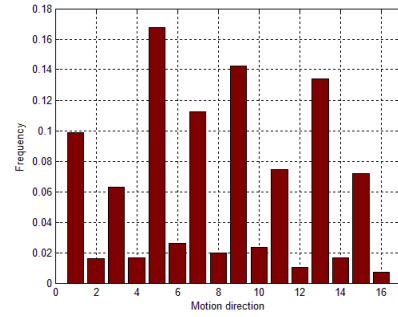
$$0 \leq W(Q, R) \leq 1$$

In principle, histogram intersection method evaluates the similarity between two histograms by taking the intersection of each corresponding bin (*i.e. taking the minimum value*). Each intersection value of two compared bins represents the amount of *moving MB* that falls within the same direction quadrant. Moreover, if every corresponding bin has similar amount of *moving MB*, the summation of all intersection value would be closer to 1. This entails that the compared videos are more likely to be similar.

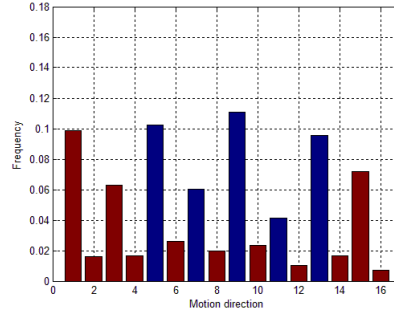
On the other hand, if the amounts of moving MBs in a number of quadrants are relatively different for two compared histograms, it is more likely that the compared videos are considered different. Consequently, the histogram intersection value is closer to 0 than 1. An illustration of histogram intersection method is shown in *figure 3.7*.



a) Histogram  $Q$



b) Histogram  $R$



c) Histogram intersection between  $Q$  and  $R$ , in which  $W(Q, R) = 0.78$

Figure 3.7: Histogram intersection method. Performing histogram intersection for a) and b) result in histogram c).

## Chapter 4 Experimental Setup

This chapter explains the experimental setups to evaluate our proposed video fingerprinting method. In section 4.1, details regarding the type of video sequence that are used in the test are given. Section 4.2 explains the required experiment to analyze our system design parameters, which are histogram bin size and frame-grouping size. Section 4.3 discusses the experiment regarding temporal misalignment between query video and database video. Section 4.4 presents the second experimental setups regarding distorted video, given that different type of distortion might change the video quality. Consequently, the generated fingerprint might also be different compared with original fingerprint. The type of distortions is selected according to common video distortions (*e.g. bitrate change, cropping, scaling*). Section 4.5 presents the experimental setup to analyze the possibility of using multiple fingerprints to obtain a better matching value. And finally, section 4.6 emphasizes on the type of measurement required in analyzing the system performance for each experiment.

### 4.1. Data set descriptions

The video dataset used in this experiment consist of 84 video clips, approximately 5 minutes each (*i.e. in total of ~7hours*). These data are part of MUSCLE-VCD (*Multimedia Understanding through Semantics, Computation and Learning - Video Copy Detection*) 2007 database [14]. Furthermore, the video clip selection ranges from different category of video clips such as documentaries, movies, black & white movies, animation, sport events, interviews and amateur videos. The information regarding the original dataset, from which the original fingerprints are extracted, are given in *table 4.1*. The programming tools utilized during the experiments are also given in *table 4.2*.

Video characteristics	Original video	Remark
<i>format</i>	MPEG-1 compression	-
<i>resolution</i>	352 x 288 pixels ( <i>CIF size</i> )	<i>Equal to 18 x 22 MB</i>
<i>frame rate</i>	25 fps	-
<i>bit rate</i>	1000 Kbps – 3000 Kbps VBR ( <i>variable bit rate</i> )	<i>Different for each video clip</i>
<i>group of picture (GOP)</i>	<b>BBIBBBBPPBBPBBP</b>	$M = 15, N = 2$
<i>frame temporal-reference</i>	Open GOP structure is used	<i>B-frame could perform inter coding by referencing frame from previous GOP structure</i>

Table 4.1: Characteristics of the original videos.

Tools	Remark
<i>mpeg_stat.exe</i>	<i>This tool is provided by Berkeley Multimedia Research Centre (BMRC) [15]. The function of this tool is to extract the motion vectors from the MPEG-1 video streams. Furthermore, this tool also extracts the video frame type, and motion vector's type per MB</i>
<i>block2specs.pl</i>	<i>This tool, also provided by BMRC, further refines the output of mpeg_stat.exe, thus makes it easier to analyze.</i>
<b>MATLAB</b>	<i>This tool is used for generating the motion fields, the fingerprint itself (i.e. motion field's direction histogram) and performing all experiments respectively, given the extracted data from the previous tools</i>
<i>ffmpeg, mencoder, Virtual Dub</i>	<i>These tools are used for applying distortion to original video clips (e.g. changing bit rate, cropping, scaling)</i>

Table 4.2: Programming tools utilized during the experiments.

## 4.2. System design parameters

The first part of the experiments is performed in order to analyze two system design parameters, which are:

### 1) Frame-grouping size

The generated histograms in this experiment are obtained using different size of frame-grouping. The motivation for this is to analyze the effect of clustering motion direction of a group of frames into single histogram. Furthermore, this experiment investigate the possible trade-offs between the fingerprinting parameters, specifically the trade-off between the size of the fingerprint and its discriminative feature. As a remark, both database and query video should always use the same frame-grouping size.

### 2) Histogram bin size

Histogram function is considered as quantization process, seeing as data are mapped into a fixed intervals of directions. Consequently, performing histogram function to a given data might lose its distinctive feature. Due to this concern, the optimal quantization factor (*i.e. histogram bin size*) should be analyzed, in order to capture the salient part of the data.

Within the experiment, the analyzed histogram bin size is always a factor of two. Therefore, if the generated histogram is again quantized (*i.e. fingerprint post-processing*) in a binary form, each bin would correspond into a single bit. Subsequently, the binarized histogram size would be a factor of two, which is considered *computer-processing* friendly. Nonetheless, the research scope does not go further in quantizing the generated histogram in a binary form since our research scope is to analyze the feasibility of using motion direction to represent a video.

### 4.3. Temporal misalignment experiment

As one of the required parameter, a video fingerprinting method only requires small part of a video (*i.e. as a query video*) to perform a correct identification. However, due to frame-grouping process that is introduced in our system, a temporal misalignment between the fingerprint of the query video and original fingerprint stored in the database might occur. An illustration is given in *figure 4.1*.

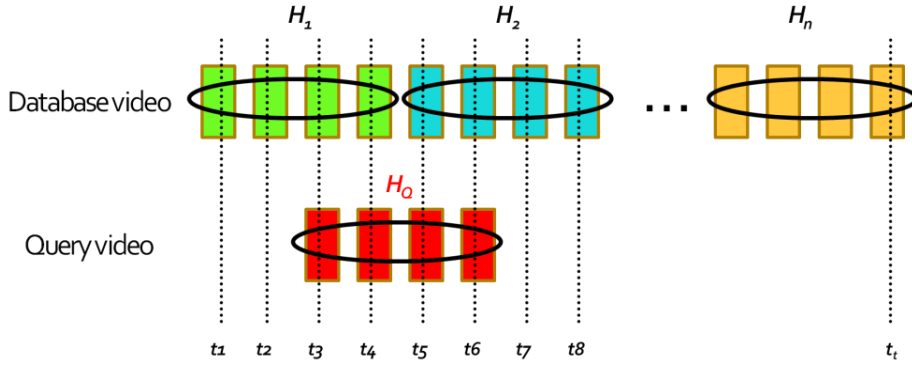


Figure 4.1: Temporal misalignment between database and query video

In *figure 4.1*, the query video's fingerprint  $H_Q$  is generated from the video frames ranging from  $t_3 - t_6$  in the temporal domain. However, the corresponding video within the database has two sequential fingerprints  $H_1$  and  $H_2$  generated from the video frames ranging from  $t_1 - t_4$  and  $t_5 - t_8$  respectively. It can be seen that there is a temporal overlapping between  $H_Q$  and ( $H_1$  and  $H_2$ ). Due to this phenomenon, a test is considered to analyze the temporal shifting effect to the matching process. The test is done by generating statistics of every similarity measurement between shifted query video and the database. Naturally, this problem would not occur if frame-grouping is not considered (*i.e. for both database and query video*).

#### 4.4. Video distortion experiment

In a digital format, video are more prone to different type of video processing (*i.e. video distortions*) where it could either affect the quality of the original video. Since video fingerprinting method should be robust against different type of distortion, this second part of the experiment is dedicated in analyzing different type of common attack that could perceptually changes the video content, either slightly or significantly. *Table 4.3* gives the general idea of each types of video distortion considered in the experiments. An example for each type of distortion is presented in *table 4.4*.

Distortion type	Remark
<b>Different bit rate</b>	<i>Decreasing bit rate (i.e. decreasing the video quality) is considered as the most common compression scheme to achieve smaller video size</i>
<b>Different GOP structure</b>	<i>Since motion field calculation is also related with GOP structure, this type of experiment is considered</i>
<b>Different bit rate &amp; GOP structure</b>	<i>This experiment is performed to analyze whether the combination of the previous tests has significant effect or not</i>
<b>Contrast adjustment (maximum contrast)</b>	<i>Video distortion is applied to differentiate objects within video frame, by changing both color and brightness of each object</i>
<b>Contrast adjustment (negative image)</b>	<i>Tonal inversion is performed for each video frame, in which light areas appear dark and vice versa</i>
<b>Brightness adjustment</b>	<i>Increasing the brightness of the entire object in video frames with the same level</i>
<b>Scaling</b>	<i>Increasing or decreasing the video resolution</i>
<b>Subtitling</b>	<i>Text is inserted, typically in the lower part of the video</i>
<b>Cropping</b>	<i>Removal of the outer part of the video frame. This processing is commonly performed if a certain aspect ratio is required</i>
<b>Spatial shifting</b>	<i>Video frames are shifted outside the border, therefore leaving empty areas inside the frames.</i>
<b>Spatial shifting (with wrapping)</b>	<i>Although spatial shifting occurs, the shifted pixels appear again in the other part of video frames, which causes spatial misalignment</i>

Table 4.3: Description for each types of video distortion that are applied













#	Distortion type	Original frame	Distorted frame
1	<i>different bit rate</i>		
2	<i>different GOP structure (same bit rate)</i>		
3	<i>different bit rate &amp; GOP structure</i>		
4	<i>contrast adjustment (maximum contrast)</i>		
5	<i>contrast adjustment (negative image)</i>		
6	<i>brightness adjustment</i>		

Table 4.4: Types of video distortion that are considered (continue to the next page)











#	Distortion type	Original frame	Distorted frame
7	<i>scaling</i>		
8	<i>subtitling</i>		
9	<i>cropping</i>		
10	<i>frame spatial shifting</i>		
11	<i>frame spatial shifting (with wrapping)</i>		

Table 4.4: Types of video distortion that are considered.

#### 4.5. Multiple fingerprints query experiment

Up to this point, all experiments only perform similarity measurement between two single fingerprints. However, it is assumed that by using more than one fingerprint, a better matching result could be obtained. In order to calculate the matching value between multiple fingerprints query, the following equation is used,



$$T = \frac{1}{G} \sum_{i=1}^G W_i$$

Where  $T$  is the matching value between groups of fingerprints,  $G$  is the number of utilized fingerprint segment and  $W$  is the histogram intersection value. Concisely, histogram intersection values are averaged to get the matching value for compared queries. As an illustration of similarity measurement with increasing fingerprint segment, consider the following figure 4.2.

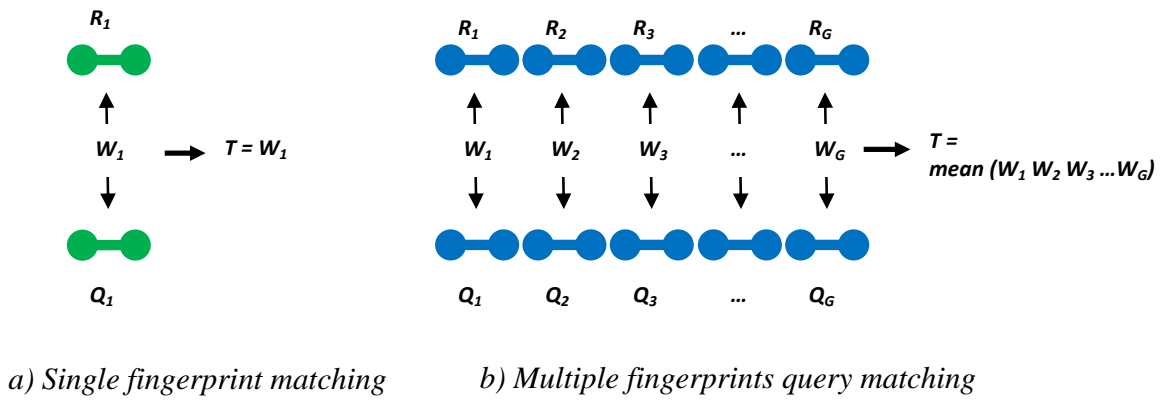


Figure 4.2: Difference between single a) and multiple fingerprints query b) matching process, where  $R$  and  $Q$  are the fingerprints for database video and query video, respectively.

Using the proposed calculation, we analyze the system performance whether the detection rate improves or not when using multiple fingerprints query. To analyze this calculation method, we perform the test given query video with temporal frame shifting. The idea is to analyze whether increasing the fingerprint size improve the system detection rate towards temporal misalignment or not.

#### 4.6. Performance measurements

Every content based identification system has to achieve reliable results in performing binary decision problem (*i.e. in this case, similar or different*). In order to evaluate the system's performance in general, extensive measurement needs to be done on a collective set of video. The generated statistics are obtained from histogram intersection value of every two compared fingerprints. Furthermore, the generated statistics can be defined into two classifications, namely intra-statistics and inter-statistics.

Intra-statistics are acquired from the histogram intersection value of two video contents, one of which is the distorted version of other. These values represent the robustness parameter of our video

fingerprint system in identifying perceptually similar data. To obtain these values, every fingerprint from distorted video is compared with every fingerprint from the original video, which is considered to be perceptually similar.

On the other hand, inter-statistics are acquired by comparing perceptually different video content. These values represent the uniqueness of the generated video fingerprint. Therefore, every generated fingerprint from the distorted video is compared with every original fingerprint, which is considered to be different. As an illustration of inter and intra-statistics calculation, consider the following *figure 4.3* and *figure 4.4* respectively.

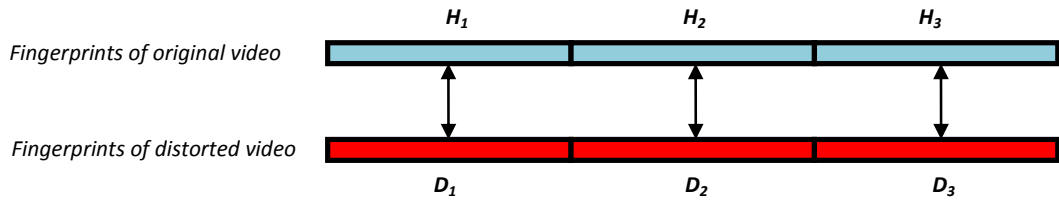


Figure 4.3: intra-statistics calculation

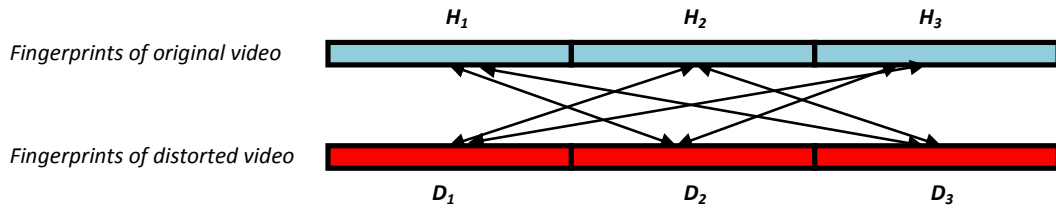


Figure 4.4: inter-statistics calculation

Both generated statistics can be considered as the probability distribution of two types of classification, namely, “two videos are similar ( $P_S$ )” or “two videos are different ( $P_D$ )”, as illustrated in *figure 4.5*. Using these probability distributions, binary decision problem can then be analyzed. As a note,  $P_S$  is expected to be closer to 1 (*i.e. more similar*) whereas  $P_D$  is closer to 0 (*i.e. more different*).

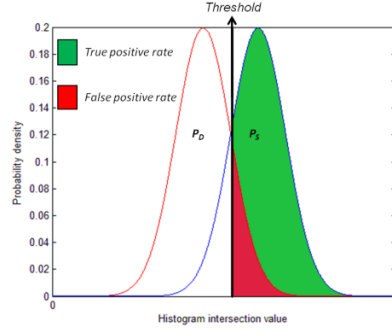


Figure 4.5: Probability distribution of  $P_D$  and  $P_S$

One of the most common evaluation metrics for analyzing binary decision problems is by using Receiver Operating Characteristic (ROC) curve. ROC curve is the graphical plot between False Positive Rate (FPR) and True Positive Rate (TPR). FPR correspond to the error probability where two different video are considered to be similar, while TPR correspond to the robustness of the video fingerprinting system in identifying perceptually similar video. ROC curve is generated by varying the discrimination threshold value along the possible range of similarity values (*i.e. histogram intersection value*), which is from 0 – 1. An illustration for a ROC curve obtained from intra and inter-statistics is shown in figure 4.6.

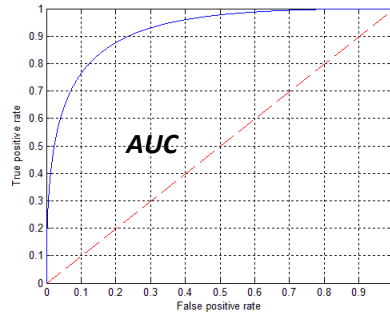


Figure 4.6: ROC curve

The shape of ROC curve shows the performance of the system in general. The closer the curve is to the upper-left corner of the axis, the better the performance is. While the diagonal line of the ROC curve corresponds to random prediction. Given this characteristic, the common evaluations metric in analyzing ROC curve is to analyze the size of area under curve (AUC). If AUC is closer to 1, the better the performance is. While if it's closer to 0.5, it means that the system behaves towards random in performing a decision. Other common method to analyze system performance is to fix one of the ROC curve axis value (*e.g. the FPR value*) and then compare the other axis value (*i.e. the TPR value*) of every other ROC curves. These evaluation metrics are used in analyzing the experiment results, which are given in the next chapter.

## Chapter 5 – Experimental Results

This chapter presents the experimental results of our proposed video fingerprinting method. The following results are given according to the experimental setup and performance measurements, explained in chapter 4. In section 5.1, we investigate the effect of our system design parameter (*i.e. histogram bin size and frame-grouping size*) towards the system performance. In addition, this section also clarifies the reasoning of discarding the *zero-motion MB*  $\langle 0, 0 \rangle$  in the histogram generation. Section 5.2 gives the result of temporal frame shifting test, in which a temporal misalignment occurs between queried and original fingerprint. Section 5.3 present the experimental results given different types of video distortion. Lastly, Section 5.4 discusses the possibility of using sequence of fingerprint queries to improve the matching performance. To simplify the discussions, the following acronyms are used within this chapter:

- ROC: Receiver Operating Characteristic
- AUC: Area under the ROC curve.
- TPR: True positive rate
- FPR: False positive rate
- $P_D$ : Probability distribution of histogram intersection value when two videos are different.
- $P_S$ : Probability distribution of histogram intersection value when two videos are similar.
- VBR: Variable bit rate
- GOP: Group of picture pattern

In addition, the term histogram and fingerprint are used interchangeably throughout the chapter.

Every experiment results in this chapter are evaluated using the AUC values of every ROC curves. However, most of the generated ROC curves are similar with each other in the overall performance (*i.e. similar AUC values*). Therefore, we also consider another approach to visualize the system performance by fixing the same FPR for every experiment and subsequently analyze the corresponding TPR. In this thesis, we consider FPR of **0.01 (1%)** as the appropriate rate. Therefore, 1 different fingerprint is incorrectly detected as perceptually similar for every 100 fingerprint comparisons.

### 5.1. Experiments on design parameters

This section is further divided into three subsections. The first subsection discusses the reasoning of excluding *zero-motion MB* in the fingerprint generation. The second subsection presents the experimental results of changing the frame-grouping size, which is the first design parameter of our

method. Consequently, the second experiment regarding our second design parameter, which is the histogram bin size, is presented in the third subsection. For consideration, this experiment only analyzes the effect of system design parameters. Therefore, the effect of video distortion towards the system performance is not considered within this section.

### 5.1.1. Zero-motion MB exclusion

As explained in the third chapter, *zero-motion MB* contains no motion direction information, since the pixels value within consecutive macro-blocks are considered stationary. In order to exemplify the reasoning for not using this MB type in the histogram generation, we present a similarity measurement between two perceptually different videos, as shown in figure 5.1.



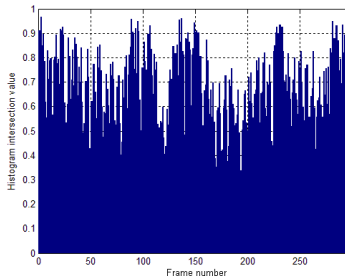
a) Video 'silent.mpg'



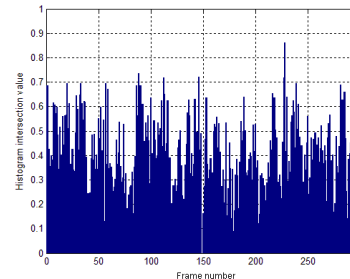
b) Video 'akiyo.mpg'

Figure 5.1: Two perceptually different video of 300 frames

These two videos have a similarity in which a person is moving given a stationary background. By analyzing visually, these two videos have similar amount of *non-moving* region. Logically, the macro-blocks within this region are considered as *zero-motion MBs*. In order to illustrate the effect of *zero-motion MB* in the similarity measurement, we generate two motion direction histograms for each video, in which we consider of using *zero-motion MB* or not. Consequently, the generated histograms of two videos are respectively compared. The comparison results are shown in the following figure 5.2.



a) With zero-motion MB



b) Without zero-motion MB

Figure 5.2: Histogram intersection values between two videos given frame-by-frame comparison with zero-motion MB a) or not b)

These results show that considering *zero-motion MB* for motion direction histogram generation can affect the similarity measurement. By using *zero-motion MB*, the histogram intersection value is closer to 1, which means the generated histograms are considered more similar. However, the videos should be considered as different (*i.e. histogram intersection value closer to 0*). On the other hand, if we discard *zero-motion MB* in the histogram generation, histogram intersection value is moving closer to 0. Therefore, these two videos are more likely to be decided as different.

This experiments shows that discarding *zero-motion MB* present a better result of similarity measurement. This is due to the effect of *zero-motion MB* towards histogram normalization. If a video contains a considerable amount of *zero-motion MB*, the *moving MB* will be less significant after the normalization compared with *zero-motion MB* (*i.e. the normalization factor is the total macro-block that is used*). Therefore, two perceptually different video with minor motion occurrence can be detected as similar. However, if we discard the *zero-motion MB*, even less *moving MB* becomes more significant to represent the video content.

### 5.1.2. Experiment on frame-grouping size

In order to analyze the effect of different frame-grouping size, we re-encode the original dataset with the characteristics of GOP structure **BBIBBP** while still maintaining the same video quality. Furthermore, we generate test result by varying the size of {5, 10, 15, 20, 25, 50, 100, 250, 500, 1000 and 1500}. For each test, both intra and inter-statistics are generated, which are used to obtain the ROC curves. Due to visualization problem, we only plot the results of frame-grouping size of {5, 25, 50, 500 and 1500}. Nevertheless, the results of other variations typically fall within the same area. The following *figure 5.3* illustrates the generated ROC curve.

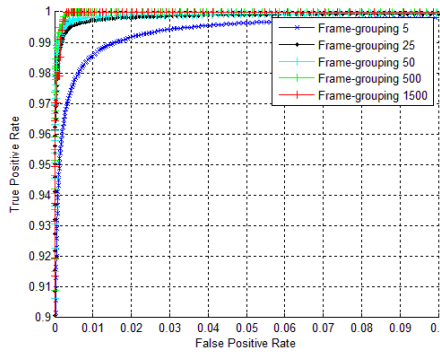


Figure 5.3: ROC curve for different frame-grouping size

As presented in this figure, we find that TPR values for each frame-grouping size are very high given  $FPR = 0.01$ . Even after increasing the frame-grouping size up until 1500 frames (*i.e.* 1 minute), we still attain unique motion direction histograms. Nevertheless, it is expected that at some point of frame-grouping size, fingerprint's uniqueness would start to decrease. As an illustration of this argumentation, consider the following *figure 5.4*, in which we plot the  $P_D$  given different sizes.

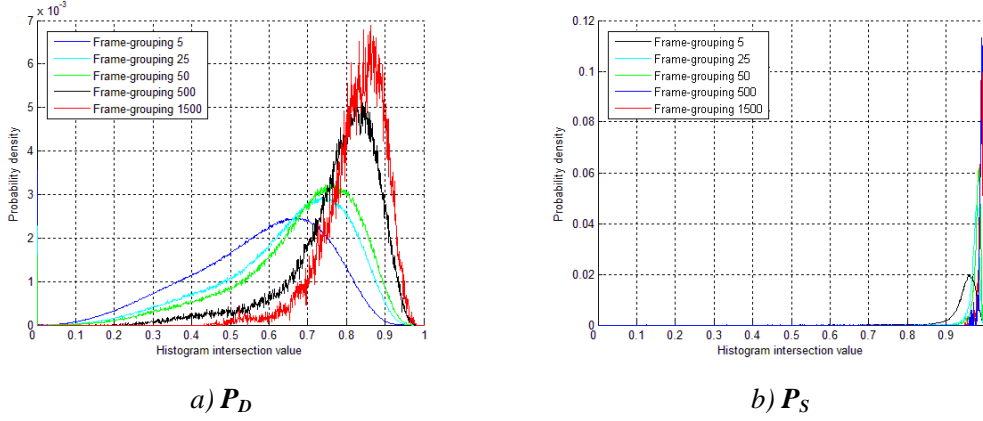


Figure 5.4: The comparison of two different videos' histogram intersection values distribution a)  $P_D$  and b)  $P_S$  given different frame-grouping size

In this figure, both probability distributions  $P_D$  and  $P_S$  is moving towards 1, with the increasing size of frame-grouping. This shows that two video fingerprints are becoming more similar, despite being similar or different in content, given a longer sequence of video frames. Consequently, we can say that the amount of motion in any video are becoming similar within a longer time frame.

To take into account, this experiment is performed without considering any quality difference between the original video and query video (*e.g.* same bitrate, no video distortions). Therefore, the amount of motion of a video can still be unique, even in a longer time frame. However, subsequent experiments shows that different type of distortions can give inferior results, given longer frame-grouping size.

In addition, the size selection also depends on the system implementation. For instance, it might be possible that the implementation requires a larger frame-grouping due to fingerprint storage limitations. Therefore, a larger group of video frames can be represented with just a single histogram. Moreover, if we decide to quantize the histogram into a binary form, we will lose more information given a larger size. Nevertheless, this research does not have any predetermined requirement for frame-grouping size. Therefore, every following experiment is always tested with different size.

### 5.1.3. Experiment on histogram bin size

In order to uniquely characterize video frame's motion, we need to quantify the motion direction sufficiently. To analyze this parameter, we vary the histogram bin size by {4, 8, 16, 32, 64, 128 and 256} and then evaluate the system performance. In this test, the re-encoded videos have the GOP structure **BBIBBP** and bitrate of 200 Kbps. To visualize the results, we select FPR of 0.01 and analyze the TPR for of each bin size test.

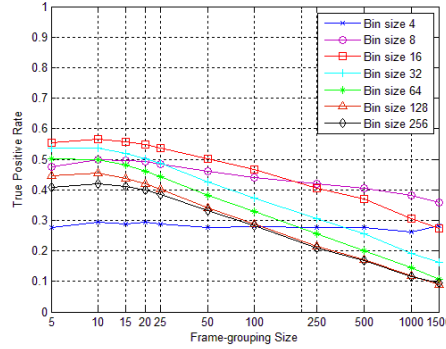


Figure 5.5: TPR values for different bin size, given different frame-grouping size

The results shown in *figure 5.5* illustrate that bin size of 16 achieve the optimum TPR compared with other bin sizes, in which TPR increases from bin size of 4 until 16 and starts decreasing afterwards. To take into account, the effect increasing the frame-grouping size is more significant given the lower bitrate constraint. Therefore, the TPR values are lower compared with the frame-grouping experiment.

To further illustrate the result of this experiment, consider the following *figure 5.6*, in which three motion direction histograms are generated with different bin sizes of {8, 16 and 32}.

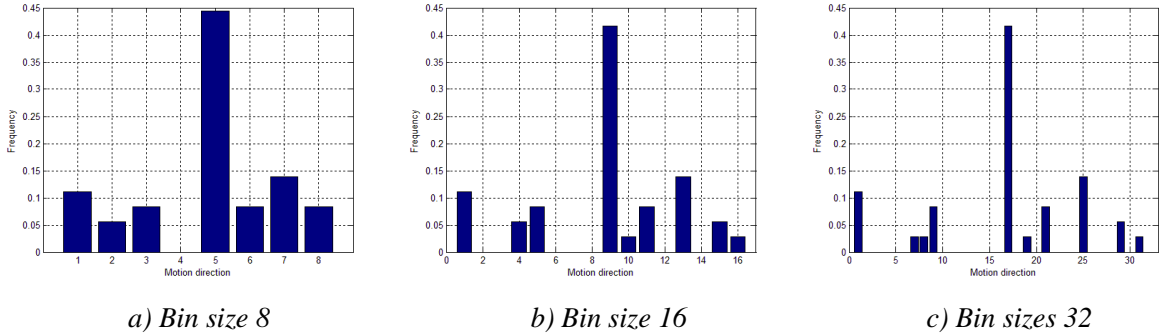


Figure 5.6: Motion direction histograms, with different bin size of a) 8, b) 16 and c) 32



By quantizing motion directions into 16 quadrants, significant motions for each quadrant can still be obtained. However, further quantizing it into 32 quadrants does not give any significant improvements. We assumed that optimum bin size of 16 is due to the effect of *ffmpeg* default motion estimation search range, in which we do not modify in any way. To provide an illustration of this argument, consider the following figure 5.7.



a) Search range with radius of 1 half-pixel      b) Search range with radius of 2 half-pixels

Figure 5.7: Motion direction quadrant according to the search range with a) divided into 4 quadrants and b) divided into 8 quadrants

These illustrations show that the optimum quadrant required for uniquely capturing each motion direction depends on motion estimation search range. Furthermore, the optimum quadrant size is four times the search range radius such that increasing the quadrant size result in trivial improvement. Due to this argumentation, it is assumed that *ffmpeg* motion estimation search range is up to 4 half-pixel displacements in every direction, which results in 16 as the optimum quadrant size. Consequently, the following experiments use histogram bin size of 16.

## 5.2. Experiment on temporal misalignment

To begin with, the test videos are re-encoded from original videos with GOP structure **BBIBBP** while still maintaining the same video quality (*i.e. same bitrate*). To analyze the effect of frame shifting, we temporally shift the query video to the left and right, in accordance with the original video. Furthermore, we consider the maximum frame shifting as half of frame-grouping size, in order to maintain enough frame correlations between query fingerprint and the original fingerprint. Finally, every shifted fingerprint is compared with the corresponding original fingerprint stored in database. A simple illustration of this test is given in figure 5.8.

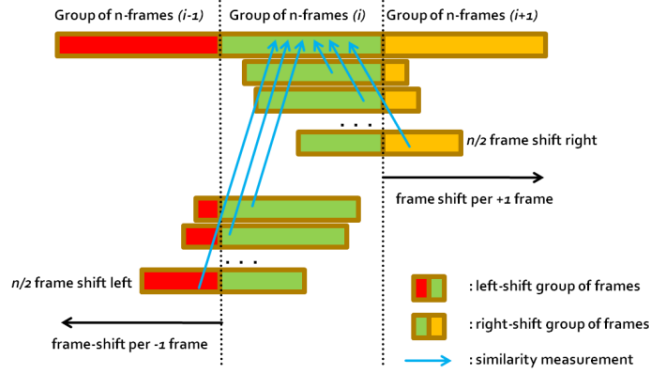
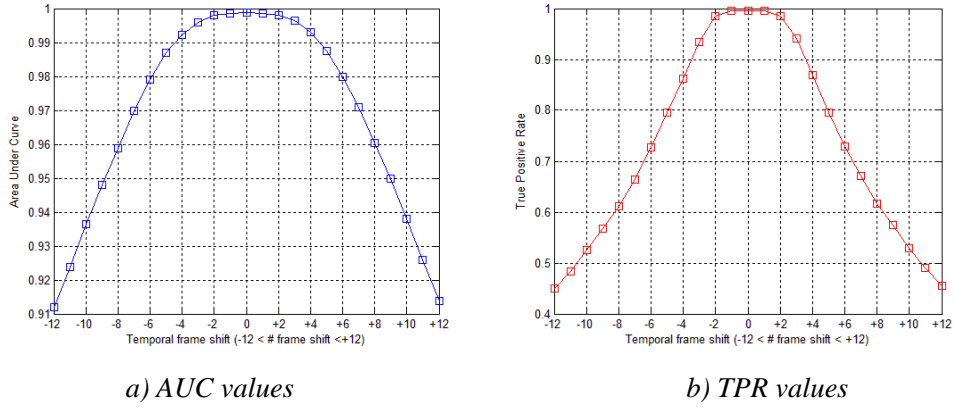


Figure 5.8: Temporal misalignment experiment procedure

In this test, we analyze the effect of temporal frame shift, given frame-grouping size 25. Considered temporal shifts are from -12 frames to the left to +12 to the right. For every shift, we generate the ROC curve, in which the AUC values and TPR values are presented in figure 5.9.



a) AUC values

b) TPR values

Figure 5.9: Temporal shift from -12 to +12 frame shifts, with frame-grouping size 25

In a real case, the amount of frame shift of a query video would not be identifiable. Therefore, we perform averaging process for intra and inter-statistics of every frame shift type and generate the *average* ROC curve. This ROC curve represents the query video, in which the temporal shift is unknown. The result of the averaging process is presented in figure 5.10. As a comparison, we also present the average ROC curve from a test using frame-grouping size 100 in this figure.

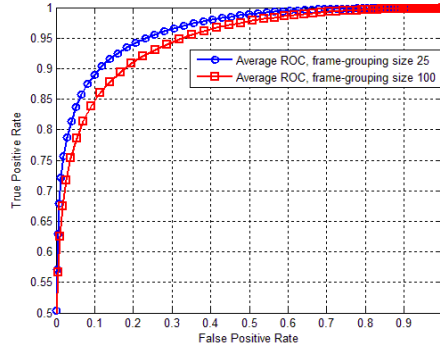


Figure 5.10: Average ROC curve of unidentified temporal shift, given two different frame-grouping sizes of 25 and 100

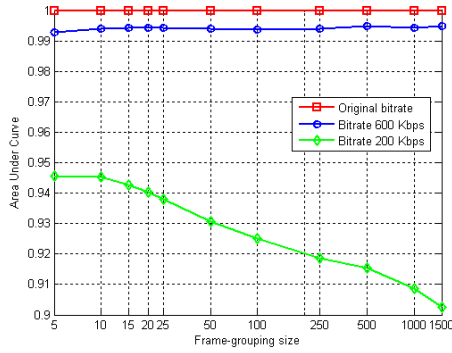
These results show that temporal frame shifting could affect the fingerprint robustness, slightly or significantly, according to the amount of shifted frames. Nevertheless, this is expected since we do not use any frame overlapping to generate the histograms (*i.e. we grouped the video frames in discrete fashion*). Overlapping frame-grouping can be performed to increase the system robustness in regards to temporal frame shifting. However, the amount of generated fingerprints significantly increases. In addition, inferior performance towards temporal misalignment is obtained as frame-grouping size increases. This is due to larger temporal misalignment between original fingerprint and query fingerprint.

### 5.3. Experiments on different types of video distortion

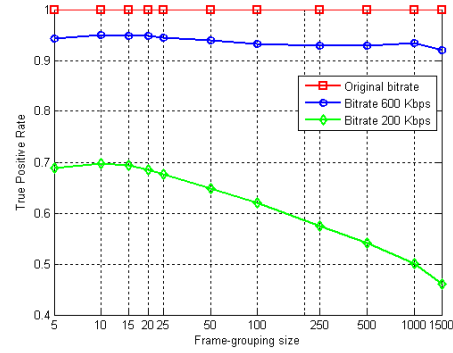
The following discussions in this section are divided according to each type of video distortion described in chapter 4. As a reminder, the original fingerprints within these experiments are generated from original videos that have GOP structure  $M = 15$ ,  $N = 2$  and  $1000\text{ Kbps} - 3000\text{ Kbps}$  VBR.

#### 5.3.1. Different bitrate

In this experiment, we re-encode the original dataset with the same GOP structure but different bitrate. The first and second test videos are encoded with 200 Kbps and 600 Kbps respectively. The following *figure 5.11* presents the results of this experiment.



a) AUC values



b) TPR values

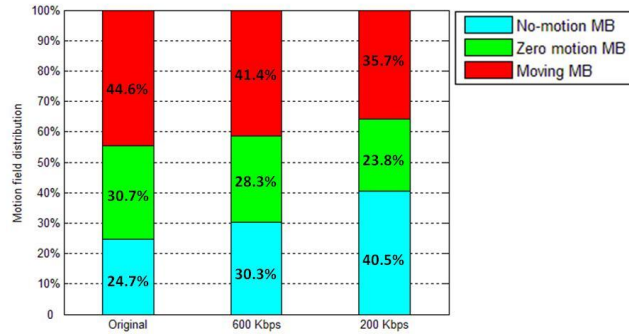
Figure 5.11: Different bitrate experiment results

These results show that the fingerprints generated from low bitrate videos are less distinctive compared with the ones generated from higher bitrate videos. This is due to the effect of bitrate constraint towards the motion estimation algorithm, in which macroblocks are skipped more frequently. Therefore, the calculated motion fields are less precise compared with the ones generated from the original video.

To further analyze the effect of bitrate constraint towards the generated motion fields, we investigate a query video of 5 minutes with different bitrates in order to identify the motion fields' distribution. The results are given in the following figure 5.12.



a) Query video

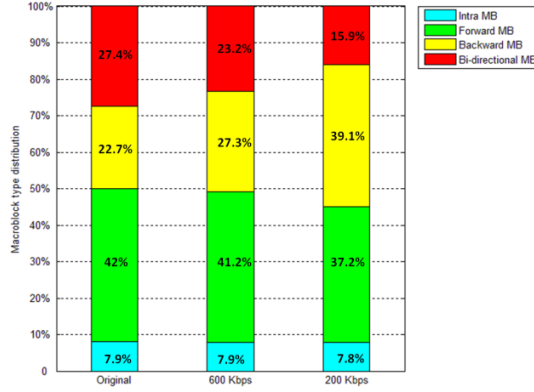


b) Motion field distribution

Figure 5.12: motion field distribution of a video sequence with different bitrates

As shown in these result, the amount of *no-motion MB* increases as the bitrate decreases. Consequently, the amount of *moving MB* becomes lesser, which affect the uniqueness of the generated fingerprints. In order to analyze the reason of increasing *no-motion MB*, we perform a simple experiment

to evaluate the macroblock prediction type distribution within the same query video (*i.e. intra, forward, backward and bi-directional MB*), given different bitrates. The results are shown in *figure 5.13*.

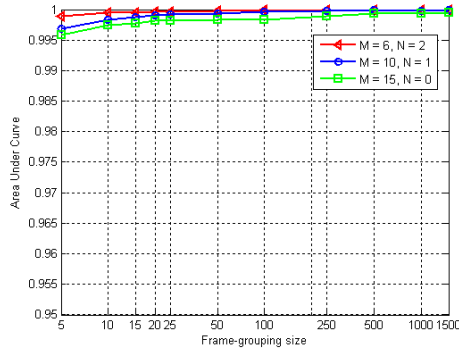


*Figure 5.13: Macroblock prediction type distribution of a video sequence with different bitrates*

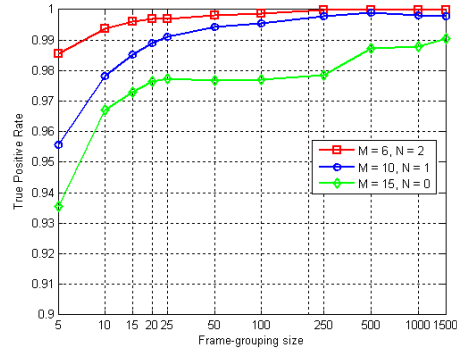
As illustrated in this figure, the distribution of macroblock prediction type changes as the bitrate decreases. It is interesting to see that the amount of *bi-directional predicted* macroblock decreases according to the decreasing bitrate. In this particular video, it shows that the encoder tend to choose only one motion vector due to bitrate constraint, rather than using two motion vectors. Therefore, the *bi-directional predicted* macroblock can become either *forward predicted* or *backward predicted* macroblock. As a consequence, these newly generated different types of macroblocks affect the calculation of the motion fields, which presumably produce more *no-motion MB*. Nevertheless, this argumentation is only based on a single query video as a case study. Therefore, a more detail investigation should be performed in order to analyze the macroblock distribution and its effect towards motion field's calculation.

### 5.3.2. Different GOP structure

In this test, we re-encode the dataset using three types of GOP structures, but still maintaining the same visual quality compared with the original video. As a reminder, GOP structure is represented with two variables  $M$  and  $N$ , in which  $M$  represent the GOP size and  $N$  represent the amount of B-frames between two reference frames (*i.e. I or P-frames*). The following *figure 5.14* presents the results of this experiment.



a) AUC values



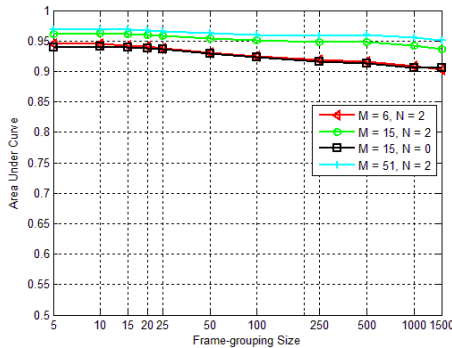
b) TPR values

Figure 5.14: Different GOP structure experiment results

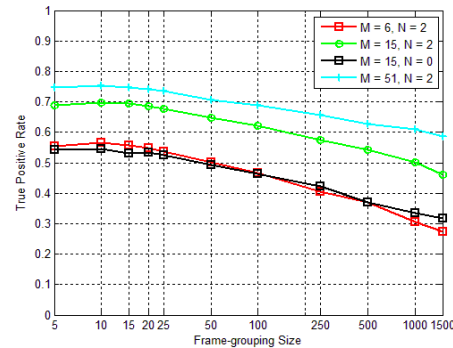
Based on these results, GOP structure has no significant effect to the identification performance if the query video has the same quality with the original video. Nevertheless, we can observe a slight difference in the TPR values of different GOP structures, in which GOP structure ( $M = 6, N = 2$ ) has the best performance compared with the other GOP structures. It appears that the ratio of frame types determines the performance of the fingerprint (*i.e. when compared with the original videos*). In these results, GOP structure ( $M=15, N=0$ ) has fairly lower performance compared with other GOP structures. Since B-frames are not used within this structure, the calculated motion fields of this structure are slightly different when compared with the original video, in which B-frames are used. Consequently, the generated histograms are also slightly different.

### 5.3.3. Different GOP structure and bitrate

In this test, we re-encode the dataset by changing both GOP structure and the bitrate. Furthermore, all re-encoded videos have the same bitrate of 200 Kbps. The following figure 5.15 presents the results of this experiment.



a) AUC values



b) TPR values

Figure 5.15: Different GOP structure and bitrate experiment results

These results show that changing both GOP structure and bitrate are still upholding the previous test's argumentation. Nevertheless, we see an interesting scenario in which the largest GOP structure ( $M=51$ ,  $N=2$ ) have the best result compared with the others. Based on these result, we can see that the amount of I-frames within the video also determines the calculated motion fields, since I-frames do not have motion vectors.

#### 5.3.4. Contrast adjustment

Contrast adjustment is the process in which the pixel's values are linearly scaled to fit within pre-defined upper and lower value limits. Therefore, the light area becomes lighter while the dark area becomes darker. In this experiment, we perform two contrast manipulations towards the original video, while still maintaining the same GOP structure and bitrate ( $M=15$ ,  $N=2$  and  $1000\text{ Kbps} - 3000\text{ Kbps VBR}$ ). The first test is to increase the contrast value to the maximum. In the second test we decrease the contrast factor to the minimum, in which the pixels values are the inverse of the original image. The following figure 5.16 illustrates the result of this experiment.

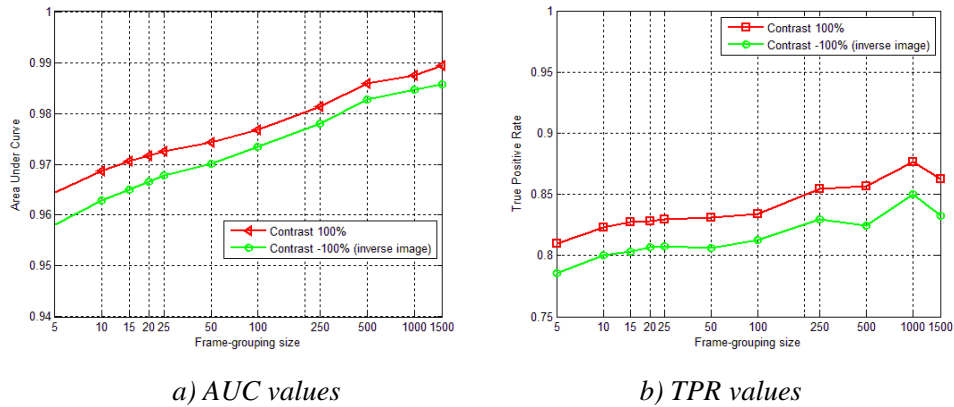
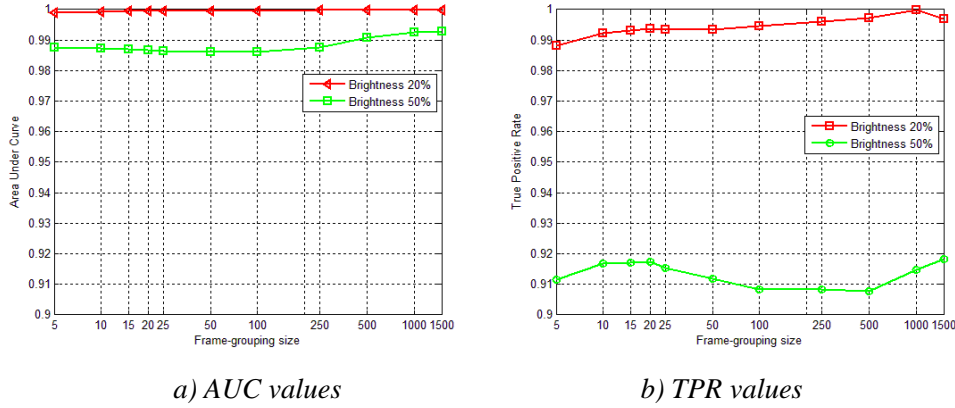


Figure 5.16: Contrast adjustment experiment results

As shown in this result, changing the video contrast affect the fingerprint uniqueness. Since pixel's intensity is changed either lighter or darker, some of the regions within the video frames lose its discriminative value (e.g. dark pixels become too dark and vice versa). Therefore, motion estimation algorithm would generate less accurate motion vectors within this area, or even skip the macroblocks altogether (e.g. motion vectors  $\langle 0,0 \rangle$ ). Due to this reason, the generated fingerprint would be less unique compared with the original fingerprint. In addition, inverting the pixel value also affect the motion estimation, since the pixels are inversely different with the original video.

### 5.3.5. Brightness adjustment

On the contrary with the previous experiment, brightness adjustment increases or decreases the overall brightness of every pixel, without any differentiation. In this experiment, we increase the video brightness, while still maintaining bitrate and GOP structure as the original video (*i.e.* GOP structure of ( $M=15, N=2$ ) and 1000 Kbps – 3000 Kbps VBR). Figure 5.17 illustrates the result of this experiment.



a) AUC values

b) TPR values

Figure 5.17: Brightness adjustment experiment results

As shown in these results, the TPR values decreases as the brightness increases. This is due to the effect of increasing the pixel luminance's value towards the motion estimation such that the generated motion vectors are less similar compared with the original one. As a remark, given an increase brightness of 100% the majority of the macroblocks are skipped MBs. This is due to the video frame pixel's saturation, in which most of the pixels are all white.

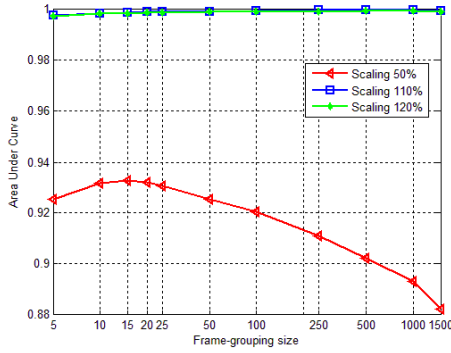
### 5.3.6. Subtitling

In this test, we insert 5 minutes long subtitle for every query video without changing the bitrate or the GOP structure. Based on our evaluation, the distortion due to subtitling is insignificant, in which the AUC and TPR values are very close to 1. The generated motion vectors are not considerably affected, since only small part of the video frame is altered due to text insertion. Therefore, the calculated motion fields do not change significantly.

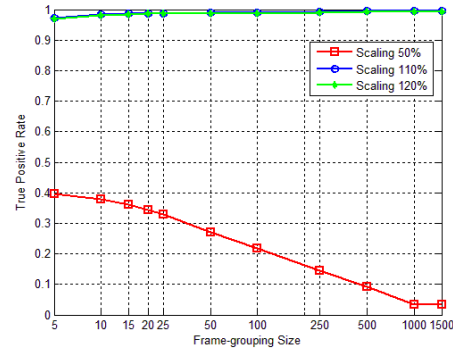
### 5.3.7. Video scaling

In this experiment, video aspect ratio is changed into smaller or larger resolution while still maintaining the same visual quality. Figure 5.18 illustrates the results of this experiment.





a) AUC values



b) TPR values

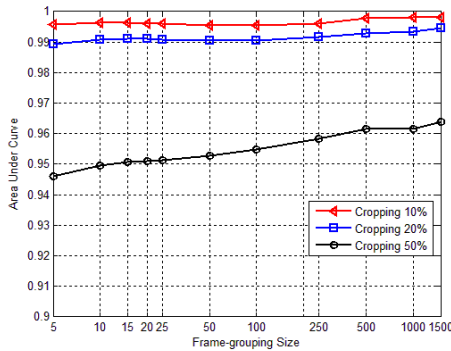
Figure 5.18: Video scaling experiment results

As shown in these figure, increasing the resolution does not give considerable distortion towards the generated fingerprints. By increasing the resolution, the encoder interpolates the pixels within the video frame, in which more pixels are generated. Nevertheless, the motion estimation algorithm still produces the same motion vector direction.

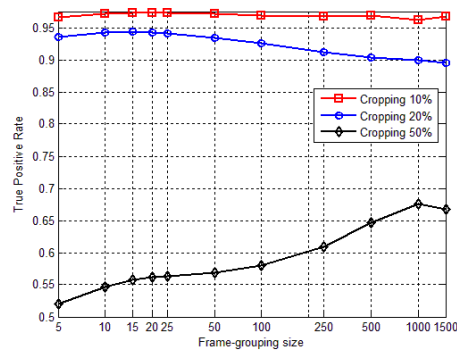
In contrast, we see a poor performance as we decrease the scaling factor to 50%. This is due to the pixels averaging given decreasing video resolution. Consequently, the motion vectors are also averaged. This affects the uniqueness of the generated motion fields' direction histogram.

### 5.3.8. Video cropping

In this experiment, certain amounts of pixels are removed from the outer region of the video frames without changing the video quality. Figure 5.19 present the results of this experiment.



a) AUC values



b) TPR values

Figure 5.19: Video cropping experiment results

As shown in these results, given the increasing cropping factor, we notice decreasing performance of the generated fingerprints. This is expected, since by performing cropping, we lose motion vector information within the cropped area. Consequently, less motion fields are obtained. Nonetheless, we see a relatively good performance given cropping factor 10% and 20%. Therefore, it can be considered that motion fields of the outer region of the video frames are less significant compared with the motion in the center of the video frames.

An interesting scenario occurs in the test given cropping factor 50%, in which an increasing TPR value is obtained as the frame-grouping size increases. This result corresponds to our consideration in which more important motion occurs within the middle part of the video frames. Therefore, given a larger *frame-grouping* size, the cumulative motion fields between original fingerprint and the query fingerprint becomes more similar.

### 5.3.9. Video frame spatial shifting

In this experiment, the query videos have the same characteristic with the original video. However, the pixels are shifted rightwards-downwards, leaving empty spaces in the pixel's original position. Furthermore, we also perform *wrapping* in which the shifted pixels are appearing again in the empty spaces. Therefore, spatial misalignments occur between original videos and query videos.

In our evaluation, the distortion caused by spatial shifting can be ignored, in which the AUC and TPR values are very close to 1. This is due to the amount of shifted pixels which are relatively insignificant (*i.e. similar with cropping*). Furthermore, the motions between consecutive frames are still similar since the same spatial shifting is performed to every video frames. Given a spatial shifting with *wrapping*, the generated motion vectors are also similar, despite the misalignment. This is expected, since we discard the spatial information of the motion vectors when we generate the histogram.

In addition, we also realized that the generated fingerprints by shifting the pixels, in which the amount is not a factor of 16, have slightly less uniqueness compared with the fingerprints, in which the shifted pixels amount is a factor of 16. This is caused by the macroblock misalignment between the query video and the original video since MPEG-1 uses fix 16x16 pixels size macroblock. Therefore, the generated motion vector could also be slightly different.

## 5.4. Experiment on multiple fingerprints query

This experiment is performed due to the assumption that using multiple fingerprints query matching would result in better identification rate. An illustration of this experiment is shown in *figure 4.2* in the previous chapter. For this experiment, we consider temporal frame shift, in which the query videos have the same bitrate but different GOP structure of ( $M=6$ ,  $N=2$ ). This experiment is performed by using frame grouping size of 25 and temporal frame shift of {4, 8 and 12}. The amount of fingerprint segments is increased from 1 to 6 segments. The following *figure 5.20* illustrates the results of the experiment.

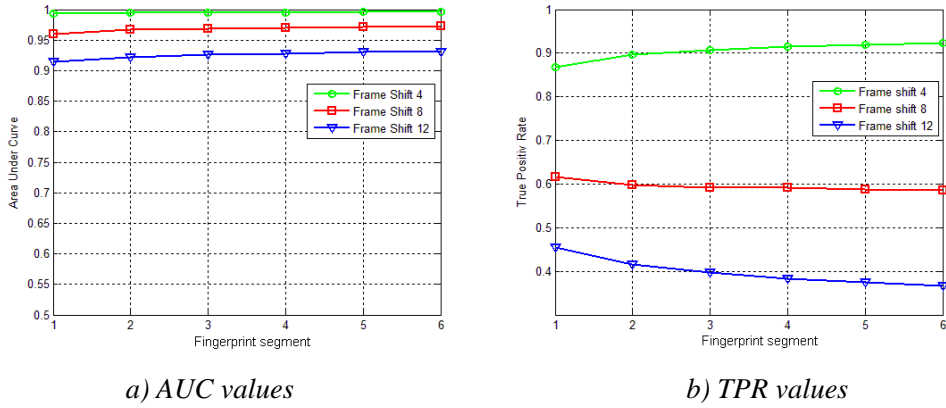


Figure 5.20: Multiple fingerprints query with temporal frame shifting experiment results

As illustrated in these results, there is no significant TPR value improvement as we increase the fingerprint size for both considered scenarios. This shows that the histogram intersection value for every compared fingerprint segment are similar with each other. Therefore, by performing averaging process to these values, we do not get a significant improvement. As a remark, the same result is also obtained even if the query video are temporally aligned (*i.e. with our without video distortions*). Due to this reason, a more improved calculation is required to analyze the matching value between multiple fingerprints.

## Chapter 6 – Discussions and open issues

### 6.1. Discussions

In this research, we investigate the feasibility of utilizing video compression parameter to generate a fingerprint. We consider utilizing motion vector, due to the fact that inter-coding is commonly performed in a compression scheme. By exploiting the motion vectors, we are able to generate the motion between adjacent video frames (*i.e. motion fields*). To obtain a unique representation feature of a video, we analyzed motion field's blocks within a number of frames (*i.e. frame-grouping*) that are moving in a certain direction (*i.e. moving MB*), and represent them in a histogram. Subsequently, the generated histogram is normalized by the total number of blocks that were used. The generated histogram is considered as the fingerprint representation of a compressed video.

The general concept of a video fingerprinting is that for any given query video (*i.e. unidentified video*), its fingerprint is compared to every original fingerprints in the database until a match is found. To perform the matching process, we use histogram intersection method, in which it evaluates whether the compared videos have the same amount of *moving MB* per direction or not. To illustrate the robustness of our method's performance, we perform an extensive experiment, in which we perform different type of distortions to the query video. Consequently, each experiment results in performance statistics that are further evaluated to determine the fingerprint's uniqueness and robustness against different types of distortion, which are common in a broadcasting channel.

Before analyzing the effect of video distortion, we investigate the effect of the system design parameters, which are frame-grouping size and histogram bin size, towards the method's performance. In regards to frame-grouping size, it shows that bitrate constraint has significant effect towards the amount of frames that can be grouped. Smaller bitrate would decrease fingerprint uniqueness, given increasing frame-grouping size. Nevertheless, the frame-grouping size should not be very large since we need to take into account the general time frame of a query video, which is presumably in a manner of seconds. Moreover, processing larger amount of frames would require longer computational time.

In regards to histogram bin size, we observe that the optimum size is correlated to the motion estimation search range of the compression scheme. However, it is unfeasible to recognize the search range of every compressed video within the broadcasting channel. Based on our results, a bin size of 16 is sufficient, since it is already represent motion search range of 4 half-pixels radius, which is considerably small. It is possible that larger bin size would produce better fingerprint. Nevertheless, larger bin size also corresponds to larger fingerprint size, which increases the computational requirement.

The main assessment of our method is to analyze its performance against different types of distortion. Based on the results, this method shows a degrading performance when the query video has lower bitrate than the original video. This is due to the effect of bitrate constraint towards the motion estimation algorithm, in which a less precise motion vector is generated (*e.g. skipped MB*). Moreover, the effect of bitrate constraint towards inter-coding also affects the amount of *moving MB*, in which lesser amount is obtained.

In regards to other types of video distortion, our proposed method shows sufficient identification performance. The results are obtained by applying rigorous distortions which are not common in a broadcasting channel (*e.g. cropping 50%, increase contrast 100%*). Therefore, it is expected that the performance of this method would be sufficient under common distortion levels.

The fingerprint uniqueness is highly correlated with the amount of pixels within a video frame. As shown in the results, the fingerprint uniqueness decreases as the amount of discarded pixels increases (*e.g. cropping, frame spatial shifting, down scaling*). This is due to the effect of the decreasing amount of macroblock within a frame. On the other hand, our method has an advantage against spatial misalignment, since we discard the spatial information during the histogram generation (*e.g. up scaling, frame spatial shifting with wrapping*).

Another important issue for our proposed method is the occurrence of temporally misaligned query video. This misalignment is due to video frames' discrete partitioning (*i.e. frame-grouping*), which is performed to generate the fingerprint. This causes compared fingerprint to be less similar, even if they are temporally close to each other. Since we cannot control both original fingerprint and query fingerprint to be temporally aligned, we propose to match multiple fingerprints query.

Multiple fingerprints query matching is performed due to the assumption that longer sequence of video frames, although misaligned, can have more motion similarity compared with shorter video sequence. Moreover, the same assumption holds for analyzing distorted query video, even if there is no temporal misalignment. Nevertheless, our computation method for analyzing multiple fingerprints query does not present any improvement towards the matching process. However, we are still certain that a more improved method could capture the similarity between compared multiple fingerprints query.

## 6.2. Open issues

In general, any proposed method can be further improved to obtain better performance. In regards to our video fingerprinting method, we observe that there are several open issues that can be addressed in the future research, which are:

- *Incorporate spatial information:* It is obvious that by generating the motion direction histogram, we discard the spatial position of every macroblock. Therefore, two videos that have same amount of motion are more likely to be similar, even if the motion field's position within the frame is spatially different. To address this issue, we propose to use *color coherence vector* method [16], since this method incorporates the pixel's (*i.e. or block's*) spatial information in the histogram generation.
- *Estimate the unidentified motion fields:* From our evaluation, it shows that the amount of *no motion MB* affects the uniqueness of the generated fingerprint. Rather than discarding this type of motion field, it might be feasible to obtain its motion by interpolating the neighboring motion fields and use it in the histogram generation.
- *Fingerprint binary quantization:* To further reduce our fingerprint redundancy, it is feasible to quantize the generated histogram into a binary sequence. This can be achieved by fixing a threshold level for every histogram, and quantize every bin value within each histogram according to the threshold. Furthermore, by having a fingerprint in a binary sequence, a more efficient matching process can be performed, such as using Hamming distance to calculate the bit error rate between compared fingerprints.
- *Implementation for other compression standard:* Our proposed method is developed in regards to MPEG-1. However, there are a number of other compression standards used in broadcasting channel such as H.263, H.264, or MPEG-2, in which each of them has different approach to perform inter-coding. Therefore, some adjustment is required in order to adapt our method with the other standards, particularly for obtaining the motion between adjacent video frames.

## Bibliography

- [1] M. AlBaqir, “*Literature Study Report on Content Based Identification*”, Information and Communication Theory (ICT) Group, Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS), Delft University of Technology, April 2008.
- [2] <http://www.youtube.com/blog?entry=on4EmafA5MA>, “Zoinks! 20 Hours of Video Uploaded Every Minute!”, 16 June 2008.
- [3] A. Hampapur and R. Bolle, “*VideoGREP: Video Copy Detection using Inverted File Indices*”, IBM Exploratory Computer Vision Group, 2001.
- [4] K. Hamon, M. Schmucker and X. Zhou, “*Histogram-based Perceptual Hashing for Minimally Changing Video Sequences*”, Proceedings of the Second International Conference on Automated Production of Cross Media Content for Multi-Channel Distribution, Pages 236-241, 2006.
- [5] Z. Li and Y. -P. Tan, “*Content-Based Video Copy Detection with Video Signature*”, IEEE International Symposium on Circuits and Systems, Pages: 4321-4324, 2006.
- [6] A. Mucedero, R. Lancini and F. Mapelli, “*A Novel Hashing Algorithm for Video Sequences*”, International Conference on Image Processing (ICIP), pages: 2239- 2242, Vol. 4, 2004.
- [7] J. Oostveen, T. Kalker and J. Haitisma, “*Feature Extraction and a Database Strategy for Video Fingerprinting*”, Proceedings of the 5th International Conference on Recent Advances in Visual Information Systems, Pages: 117 – 128, 2002.
- [8] C. de Roover, C. De Vleeschouwer, F. Lefebvre and B. Macq, “*Robust Image Hashing Based On Radial Variance of Pixels*”, IEEE International Conference on Image Processing (ICIP), Pages: 77 - 80, Vol. 3, 2005.
- [9] A. Massoudi, F. Lefebvre, C.-H Demarty, L. Oisel, B. Chupeau, “*A Video Fingerprint Based on Visual Digest And Local Fingerprints*”, IEEE International Conference on Image Processing, Pages: 2297-2300, 2006.
- [10] B. Senechal, D. Pellerin, L. Basacier, I. Simand and S. Bres, “*Audio, Video and Audio-visual Signatures for Short Video Clip Detection: Experiments on Trecvid2003*”, IEEE International Conference on Multimedia and Expo (ICME), Pages: 221-224, 2005.
- [11] A. Mikhalev *et. al.*, “*Video fingerprint structure, database construction and search algorithms*”, Direct Video & Audio Content Search Engine (DIVAS) project, Deliverable number D 4.2, February 2008.
- [12] J. Gilvarry, “*Calculation of Motion Using Motion Vectors Extracted from an MPEG Stream*”, School of Computer Applications, Dublin City University, September 1999.
- [13] M. J. Swain and D. H. Ballard, “*Color Indexing*”, International Journal of Computer Vision, 7:1, Pages: 11-32, June 1991.

- [14] <http://www-rocq.inria.fr/imedia/civr-bench/benchMuscle.html>, “Video Copy Detection Evaluation Showcase”, 16 June 2008.
- [15] <http://bmrc.berkeley.edu/frame/research/mpeg>, “Berkeley MPEG tools”, Berkeley Multimedia Research Center (BMRC), 16 June 2008.
- [16] G. Pass, R. Zabih and J. Miller, “*Comparing Images Using Color Coherence Vectors*”, Computer Science Department, Cornell University, 1997.
- [17] L. Bouarfa, “*Research Assignment on Video Fingerprinting*”, Information and Communication Theory (ICT) Group, Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS), Delft University of Technology, December 2006.
- [18] B. G. Haskell, A. Puri and A. N. Netravali, “*Digital Video: An Introduction to MPEG-2*”, Chapman & Hall, 1997.