PERCEPTUAL HASHING FOR WAVELET-BASED SCALABLY-CODED VIDEO

A

Thesis Submitted

in Partial Fulfilment of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

By

NAVAJIT SAIKIA

to the

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI

GUWAHATI - 781039, INDIA

JULY, 2010

PERCEPTUAL HASHING FOR WAVELET-BASED SCALABLY-CODED VIDEO

Navajit Saikia

Certificate

This is to certify that the thesis entitled "**PERCEPTUAL HASHING FOR WAVELET-BASED SCALABLY-CODED VIDEO**", submitted by **Navajit Saikia** (02610202), a research scholar in the *Department of Electronics and Communication Engineering, Indian Institute of Technology Guwahati*, for the award of the degree of **Doctor of Philosophy**, is a record of an original research work carried out by him under my supervision and guidance. The thesis has fulfilled all requirements as per the regulations of the institute and in my opinion has reached the standard needed for submission. The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

Dated:
Guwahati.

Prof. Prabin K. Bora
Deptt. of Electronics and Communication Engg.
Indian Institute of Technology Guwahati
Guwahati - 781039, India.

This work is dedicated to

MAHESHA and HIS FAMILY

Acknowledgements

I consider myself to be very fortunate for getting opportunity to study at the Indian Institute of Technology Guwahati, particularly at the Department of Electronics and Communication Engineering. I shall always cherish the academic ambience and the cordial atmosphere in the Department.

I am indebted to my Supervisor Prof. P. K. Bora. It is a great privilege to express my deepest gratitude for his valuable guidance and all kinds of support. I am also very thankful for the precious time that he has spent with me on this thesis. The various values that I have learnt from him shall remain a source of inspiration to me.

Prof. S. Dandapat, the Chairman of my Doctoral Committee, has been always a source of encouragement to me. I acknowledge my sincere gratitude for his valuable suggestions and helps. I would like to thank the present members of the Doctoral Committee, Dr. R. Bhattacharjee and Dr. K. Karthik for their valuable suggestions. I am also thankful to Dr. D. Ghose and Dr. J. S. Sahambi for their suggestions during their short tenures on the Doctoral Committee.

I specially thank Prof. S. Majhi, Head, Prof. A. Mahanta, Prof. A. K. Gogoi, Dr. S. R. M. Prasanna, Dr. A. Rajesh and other Faculty members of the Department for their support and constant encouragement. I fondly acknowledge the helps from Mr. Sanjib Das and Mr. L. N. Sarma during the entire course of this work. I am grateful to the all present and past Staff members of the Department.

It is an opportunity for me to thank the Administration of the Assam Engineering College, Guwahati, for allowing me to carry out this work. My special thanks goes to Prof. A. Nath, Prof. P. K. Brahma, Prof. B. D. Saikia, Prof. A. K. Kalita, Ms. Amrita Ganguly, and Ms. R. Borgohain for their encouragement and support. I am also thankful to Mr. Gokul Ch. Sarma for his endless help and moral support. My friends, Mr. Gunajit Kalita and Mr. Rituraj Phukan, are always my sources of inspiration. I would like to extend my heartfelt thanks to them.

"There is no friendship, no love, like that of the parent for the child." I thank my mother and late father for their unconditional love, understanding and support throughout my life. Their guidance and unlimited sacrifices are the reasons for what and where I am now.

I am also thankful to the family of my Supervisor, Ms. Juhi and the members of my family for their encouragement. My friends at IITG, K. Manglem Singh, Vinod P, Diganta K. Gogoi, Senthil Raja, D. Senthilkumar, Josephine S., Bhabesh Deka, S. S. Karthikeyan, Kandarpa K. Sarma, Rupaban Subadar and others, deserve my sincere thanks for their supports.

(Navajit Saikia)

Abstract

A *perceptual hash function* for video extracts a fixed-length binary string called the *perceptual hash* on the basis of the *perceptual content* of the video. Besides being sensitive to the *content differences* in videos, a perceptual hash function should be robust against the *content-preserving operations* on the videos. Recent developments in the field of *scalable video coding* (SVC) demands the robustness of the perceptual hash against the scalability features of SVC. The $3D$ *discrete wavelet transform* (3D-DWT) is a way of achieving scalable coding, wherein the inherent multi-resolution structure of the 3D-DWT is exploited. This thesis deals with content-based representation and hashing of video using the 3D-DWT for the use in the *wavelet-based SVC* (WSVC).

This thesis first considers extracting *representative frames* for video using the 3D-DWT. It examines the representation of the content of a video at the *group-of-frames* (GOF) level by the bands of the 3D-DWT decomposition. The spatio-temporal low-pass band at the full level of temporal and an intermediate level of spatial decomposition of a GOF is used for representing the content of the GOF. Experimental results show the effectiveness of the band in representing the content of the GOF.

Two perceptual hash functions are extracted from the perceptually-representative spatio-temporal low-pass band. For this purpose, the band is divided into *perceptual blocks* that are sensitive to local contents of the GOF. The first hash function derives a hash of the GOF by binarising the wavelet coefficients in each perceptual block. The similarity between two GOFs is measured in terms of the maximum Hamming distance between the hashes of the corresponding perceptual blocks. Experimental results show that the hash function is robust against the scalability features of WSVC and other content-preserving operations, and sensitive to content differences at the frame and GOF levels. The hash function has limitations of a large hash size and weak confusion and diffusion properties.

The second hash function computes a compact hash by binarising the forward and backward cumulative averages of the local means of the perceptual blocks in the spatio-temporal low-pass band. Experimental results show the robustness of the hash functions against the scalability features of WSVC and other content-preserving operations, and the sensitivity to the content differences at the frame and GOF levels. This hash function is shown to have good diffusion and confusion properties.

Contents

List of Figures

List of Tables

List of Acronyms

3D-DCT	Three-dimensional discrete cosine transform
3D-DWT	Three-dimensional discrete wavelet transform
ART	Angular radial transformation
AVC	Advance video coding
AWGN	Additive white Gaussian noise
CBR	Compression bit-rate
CDF	Cumulative Distribution Function
CIF	Common intermediate format
DCT	Discrete cosine transform
DWT	Discrete Wavelet transform
EBCOT	Embedded block coding with optimized truncation
ECC	Error correction coding
EMDC	Embedded morphological dilation coding
EZBC	Embedded zero block coding
EZW	Embedded zerotree wavelet
FIR	Finite impulse response
fps	Frames per second
GOF	Group of frames
GOP	Group of pictures
HH	High-high band of spatial wavelet decomposition
HL	High-low band of spatial wavelet decomposition
IEC	International Electrotechnical Commission
ISO	International Organization for standardization
ITU-T	International Telecommunication Union - Telecommunication standardization sector
JPEG	Joint Picture Experts Group

JVT	Joint Video Team
LABR	Lowest authentication bit-rate
LH	Low-high band of spatial wavelet decomposition
LL	Low-low band of spatial wavelet decomposition
MAC	Message authentication code
MD	Message digest
MPEG	Moving Picture Experts Group
MSB	Most significant bit
PI	Performance index
PKCS	Public key crypto-system
PMF	Probability mass function
QCIF	Quarter common intermediate format
RASH	Radial hashing
RAV	Radial variance
SHA	Secure hash algorithm
SNR	Signal-to-noise ratio
SPIHT	Set partitioning in hierarchical trees
SVC	Scalable video coding
TIRI	Temporally informative representative image
VLC	Variable length coding
WSVC	Wavelet-based scalable video coding

List of Symbols

a^i	i^{th} hash bit
B_k^l	l^{th} block in the k^{th} frame
$b^{l,q}$	q^{th} wavelet coefficient in the l^{th} perceptual block of a frame
C_1	Number of content-wise similar GOF pairs declared as similar
C_2	Number of similar GOF pairs under test
C_3	Number of content-wise dissimilar GOF pairs declared as dissimilar
C_4	Number of dissimilar GOF pairs under test
$D_{k,k-1}$	Disparity between two frames f_k and f_{k-1}
d	Hamming distance, distance metric
d'	Number of 1's or 0's in disagreement in hashes derived by median thresholding
d_k^l	Hamming distance between the l^{th} perceptual blocks in the k^{th} representative frames of two GOFs
$d_{x,y}^l$	Hamming Distance between the l^{th} perceptual blocks of two GOFs G_x and G_y
$E_{x,y}$	Differential energy between two frames f_x and f_y
$F_d(q)$	CDF of d_k^l at a point q
$F_\gamma(q)$	CDF of γ at a point q
f_k	k^{th} frame of a GOF / video segment / video shot
$\tilde{f}$	Frame after random shuffling of the perceptual blocks
$\overline{f}_k$	k^{th} frame in a normalised GOF / video segment / video shot
f_k^{tuL}	k^{th} frame in the temporal wavelet band tuL
$f_k^{tuL-svLL}$	k^{th} frame in the spatio-temporal wavelet band $tuL - svLL$
f^{key}	Key frame of a video shot
f_k^{rep}	k^{th} representative frame of a GOF
$\hat{f}_k^{rep}$	Binary version of f_k^{rep}
G	Group-of-frames
$\widetilde{G}$	High-pass wavelet decomposition filter

$\widetilde{g}$	Impulse response of the filter $\widetilde{G}$
$\widetilde{H}$	Low-pass wavelet decomposition filter
$\widetilde{h}$	Impulse response of the filter $\widetilde{H}$
h	Hash function
$h(m)$	Hash of a message m
$h(m, K)$	Hash of a message m using a secret key K
$h(V)$	Perceptual hash of the video V
$h(V, K)$	Perceptual hash of the video V using the secret key K
K	Secret key
K_{pr}	Private key
K_{pu}	Public key
k	Frame index
M	Number of blocks in a frame
m	Messages
max	Maximisation operator
med_b	Median of backward cumulative averages
med_f	Median of forward cumulative averages
N	Number of segments / GOFs in a video
$N_1 \times N_2$	Dimension of a video frame
N_3	Number of frames in a GOF / video segment / video shot
$\overline{N}_1 \times \overline{N}_2$	Dimension of a frame in a normalised GOF / video segment / video shot
$\overline{N}_3$	Number of frames in a normalised GOF / video segment / video shot
(n_1, n_2)	Pixel location in a frame
P	Probability
$P_d(l)$	PMF of a Hamming distance l between two hashes
PI	Performance index
$p \times p$	Size of a perceptual block
Q	Number of representative frames for a GOF
R_1	Representation performance for content similarity
R_2	Representation performance for content dissimilarity
S	Similarity metric
T	Threshold

tuL	Temporal low-pass band at the u^{th} level of wavelet decomposition
tuH	Temporal high-pass band at the u^{th} level of wavelet decomposition
$tuL - svLL$	Spatio-temporal low-pass band at the u^{th} level of temporal and v^{th} level of spatial wavelet decompositions
u	Number of temporal decomposition level
$\hat{u}$	Full level of temporal decomposition
V	Video
V_j	j^{th} segment of the video V
V_{query}	Query video segment
$\overline{V}_j$	Nomalised version of V_j
v	Number of spatial decomposition level
W^l	Block obtained by symmetrically augmenting the perceptual block B^l
α_k	k^{th} weight in a weighted sum
β_k^l	l^{th} value of the histogram of f_k
γ	Maximum of the Hamming distances between the corresponding perceptual blocks of two GOFs
δ	Delta function
$\varepsilon_{x,y}^l$	Difference energy of the l^{th} blocks in f_x and f_y
η_{cp}	Distortion due to a content-preserving operation
λ_k^l	$\mu_k^l - \mu_k^{l-1}$
μ	Mean of spatio-temporal low-pass band
μ_d	Mean of the Hamming distances between the hashes of distinct GOFs
μ_b^l	Backward cumulative average of the l^{th} perceptual block in a frame
μ_f^l	Forward cumulative average of the l^{th} perceptual block in a frame
μ_k^l	Mean of l^{th} perceptual block of the k^{th} frame
μ_γ	Mean of the maximum Hamming distances of distinct GOFs
σ^2	Variance of AWGN
σ_d^2	Variance of the Hamming distances between the hashes of distinct GOFs
ℓ_1	ℓ_1 metric
$\oplus$	Binary XOR operation
$\lceil \cdot \rceil$	Rounds a number to the nearest larger integer
$\|\|$	Concatenation operator

Chapter 1

Introduction

The growth of multimedia and storage technologies has brought among others two new challenges to researchers in image and video processing. First, finding an effective and efficient way to represent images and videos. Second, finding an effective and efficient solution to verify the integrity of the *perceptual content* (or *content*s) and the authenticity of images and videos. Video has both spatial and temporal dimensions, and hence the solutions to these problems should consider the spatio-temporal content of a video at a time. In the solution to the first problem, *representative frame*s based on the spatio-temporal content of the video may be extracted for representing the video. The *perceptual hash function*s provide a solution to the second problem. This thesis addresses the problems of representing and perceptual hashing of video.

1.1 Cryptographic Hash Function

A *cryptographic hash function* h is a mathematical function that converts a variable-size input message m into a fixed-size output $h(m)$ called the *hash* or the *message digest* of m [1], [2]. The message is usually a bit string of arbitrary length and the hash is a bit string of fixed length. The hash of the message is a fingerprint or a summary of the message. The basic requirements of a cryptographic hash function are summarised below.

i. *Input/output lengths*: m can be of any length and $h(m)$ has a fixed length.

ii. *Computational simplicity*: For any m, computation of $h(m)$ should be relatively easy.

iii. *Uniqueness*: $h(m)$ should be deterministic. That is, it should depend only on m.

iv. *One-wayness*: Given $h(m)$, it should be computationally infeasible to find m.

v. *Collision resistance*: It should be computationally infeasible to find any two messages m and m' such that $h(m)=h(m')$. When the hash function results in the same hash for two distinct messages, the hashes *collide*.

Given a hash function, the space constituted by the all possible hashes is called the *hash space* of the function. The number of points in the hash space depends on the hash length: the fewer the number of bits per hash is, the fewer is the number of possible hashes or points in the hash space. As the hashes are shorter than the message lengths, multiple messages may yield one hash. For minimising collisions, the hash function should map messages to the hashes as evenly as possible. That is, the distribution of the hashes should be uniform. Two of the popular hash functions, the *message digest version* 5 (MD5) and the *secure hash algorithm* 1 (SHA-1) [2], produce hashes of lengths 128 bits and 160 bits respectively. One requires to try approximately 2^{128} and 2^{160} messages respectively for finding a message that yields a particular hash, or approximately 2^{64} and 2^{80} messages respectively for finding two messages that yield a particular hash [2]. As the searching of 2^{64} messages is considered exhaustive with the current state-of-the-art computing, it is infeasible to find two messages in practice with the same hash value.

Note that the cryptographic hash functions as defined above do not involve the use of any key and they do not themselves provide hash security [2]. Hash security is important when a hash function is used to authenticate a message. It is achieved by using of a *secret key* K as another input parameter to the hash function for imparting randomness in hash computation. A secured hash $h(m, K)$ thus derived is often called a *keyed hash* [2].

The traditional cryptographic hash functions are not suitable for hashing of image or video data. For example, digital images and videos undergo various signal-processing operations that do not change their perceptual contents. Therefore, the hashing of images or video sequences that look the same to the human eye should derive similar hashes. The cryptographic hash functions cannot meet these requirements as they produce completely different hashes even when one bit is flipped in the data. By noting this deficiency, one arrives at the notion of the perceptual hash functions which are sensitive to the differences in the perceived contents only.

1.2 Perceptual Hash Function

Unlike text, image and video have a special characteristic: the *perceptual meaning* or the understanding of the content of the image or video by a user. A perceptual hash function derives a short fixed-length *perceptual hash* of an image or a video based on the perceptual content of the image or video. Image and video differ in one respect: the presence of the time axis in video. Therefore, apart from the requirements of a perceptual hash function for image, a perceptual hash function for video has also to consider the temporal information in the content.

Consider a perceptual hash function h that maps a video V into a small-length bit-string $h(V)$. Operations like frame dropping, frame resizing, requantization, recompression, brightness and contrast modifications, etc., on V change the bit representation of V without affecting the content of V. While h should be capable of discriminating dissimilar videos, it should also be *robust* against these *content-preserving operations*. Thus, *robustness* and *sensitivity* are two essential criteria for the perceptual hash functions in general [3], [4], [5], [6], [7].

Perceptual hash functions find use in applications like indexing and identification of video in databases, video content authentication, video copy detection, identification of video segments in commercial broadcasts, etc. As $h(V)$ is a deterministic function, an attacker may exploit this deterministic nature to deceive the authentication or copy-detection system. In a video authentication system, an attacker may intercept the transmission, maliciously modify the content of the video such that the hash does not change and then transmit the modified video. The user of the video being unaware of this malicious modification will wrongly accept the video to be authentic. In a copy-detection system, a pirate can copy the video and add perceptually-insignificant modification to change the hash without being caught. Therefore, a perceptual hash function for these applications includes a secret key for the protection of the hash. When the secret key K is used, it can be concealed from the attacker or pirate, and the malicious activities can be prevented. The hash security may not be always an issue of concern when a hash function is used for identifying video in databases [8]. A perceptual hash function without a key suffices unless a database is a secured one.

1.2.1 Measure of Content Similarity

Consider two videos V_x and V_y. Let $h(V_x, K)$ and $h(V_y, K)$ be their respective perceptual hashes. For comparing $h(V_x, K)$ and $h(V_y, K)$, a suitable metric $d(h(V_x, K), h(V_y, K))$ can be used. For example, the Hamming distance [9] and the generalized Hausdorff distance [10], [11] are used for

comparing hashes. To measure content similarity, a threshold T on d is introduced. The videos V_x and V_y are content-wise similar if

$$d\left(h\left(V_x, K\right), h\left(V_y, K\right)\right) \leq T, \tag{1.1}$$

and content-wise dissimilar if

$$d\left(h\left(V_x, K\right), h\left(V_y, K\right)\right) > T. \tag{1.2}$$

1.2.2 Desirable Properties of a Perceptual Hash Function

Let $h(V, K)$ represent the hash of the video V using the secret key K. The desirable properties of $h(V, K)$ are enumerated below.

Property 1: *Computational simplicity*: Evaluation of $h(V, K)$ should be computationally simple.

Property 2: *Uniqueness*: $h(V, K)$ should be a deterministic function of V.

Property 3: *One-wayness*: $h(V, K)$ should be one-way. That is, it should be computationally exhaustive to infer the content of V from $h(V, K)$.

Property 4: *Robustness*: $h(V, K)$ should be robust against the content-preserving operations. Let P represent the probability and η_{cp} be the distortion due to a content-preserving operation on V. The requirement for the robustness is: $P(d(h(V + \eta_{cp}, K), h(V, K)) \leq T) \approx 1$, where T is a suitable threshold.

Property 5: *Diffusion*: This property represents the sensitivity of the hash function to the *content differences* in dissimilar videos [12]. For two dissimilar videos V_x and V_y, this property requires that $P(d(h(V_x, K), h(V_y, K) \gg T) \approx 1$.

The diffusion property represents the collision resistance of $h(V, K)$. It should be practically infeasible to find two dissimilar videos with the same hash.

Property 6: *Localisation of content differences*: $h(V, K)$ should be capable of localising the differences in the contents of dissimilar videos.

Property 7: *Confusion*: This property corresponds to the complexity in the relationship between the secret key and the hash [12]. For good confusion property, two distinct keys K_1 and K_2

for the same video V should result in sufficiently distinct hashes. In other words, it is required that $P(dist(h(V,K_1),h(V,K_2)) \gg T) \approx 1$.

The inference of these properties is that $h(V,K)$ should provide an effective and efficient access to the content of the video V.

When one video is an attacked version of the original, **Property 5** represents the fragility of $h(V,K)$ against the attack. The fragility requirement is different for the content-authentication and copy-detection applications. While the former require $h(V,K)$ to be fragile against the minutest malicious modifications, the later requires it to be robust even against major manipulations. In the following, video identification and video authentication are presented in brief.

1.2.3 Video Identification

The quality of a video identification tool depends on how efficiently the tool can identify video in archives or video segments in broadcasts. Videos in a database are tagged with the respective hashes extracted from their contents. When a user supplies a short video segment (often called a *query segment* [13]), a video identification system identifies and delivers the video from which the query segment originates [8], [13], [14], [15], [16]. The system computes the hash of the query segment and compares it with those of the segments in each video in the database. It finds the similar segment in the database based on a similarity criterion and delivers the video containing the query segment.

The block diagram in Figure 1.1 shows the archival and identification steps in a video identification system. In the archival step, an input video V is divided into N non-overlapping segments $V_1, V_2, \cdots, V_N$. A perceptual hash function h generates a binary hash of each segment independently. Let $h(V_j,K)$ represent the hash of the j^{th} segment V_j. The video is then stored with the hashes of the segments as its index. In the identification step, given a query segment V_{query}, the hash $h(V_{query},K)$ is computed and compared with the hashes of the segments in the database by means of a distance measure. For example, the similarity between $h(V_j,K)$ and $h(V_{query},K)$ is decided according to:

$$\begin{aligned} d\left(h\left(V_j,K\right),h\left(V_{query},K\right)\right) \leq T &\quad : \quad V_j \text{ and } V_{query} \text{ similar, and} \\ d\left(h\left(V_j,K\right),h\left(V_{query},K\right)\right) > T &\quad : \quad V_j \text{ and } V_{query} \text{ dissimilar,} \end{aligned} \tag{1.3}$$

where d is a distance metric and T is a threshold chosen suitably. When a segment in the database is found similar, the corresponding video is identified.

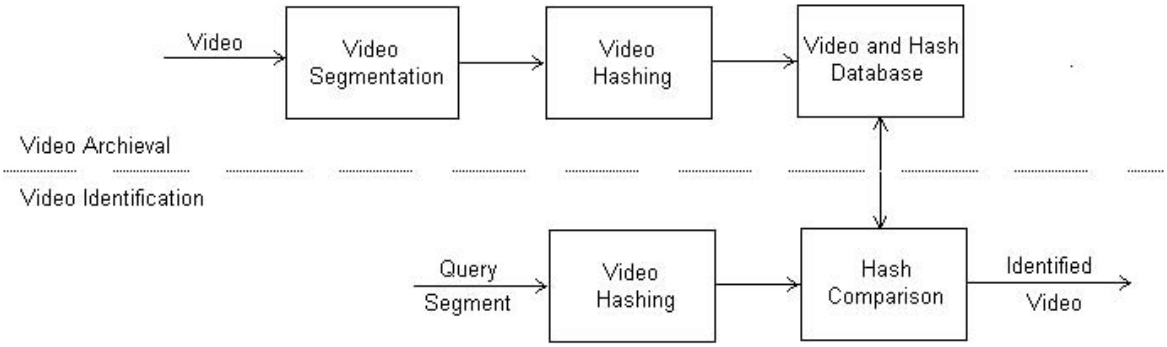


Fig. 1.1: Block-diagram representation of the archival and identification steps in a video identification system

1.2.4 Video Authentication

Powerful softwares for video manipulation have made it very easy to copy and modify videos. Consequently, video authentication or the verification of the content integrity and the origin of video has been a subject of research interest in recent times. A video authentication system enables the originator of a video to provide the receiver with means by which the origin of the video can be authenticated and the receiver can also verify that the content of the video has not been modified. The video authentication systems are primarily of two types: *watermarking* based and *digital-signature* based [17]. Watermarking embeds the security information throughout the visual data in a manner that does not impede the normal use of the data. On the other hand, a content-based digital signature scheme uses the *public-key cryptosystem* (PKCS) enabling the originator of a video to generate a signature by encipherment of a hash representing the content of the video. Although the digital signature schemes suffer from bit overhead and bandwidth requirement, they are useful in many applications. They are preferred to the watermarking schemes because of

i. no limitation on authentication data,

ii. no modification of video data,

iii. their suitability for the use in the public-key based authentication systems, and

iv. their immediate usability with all existing content [18].

A perceptual hash function is the core component of a digital signature scheme. It should immaculately distinguish the content-preserving operations from the all-possible malicious modifications on videos. Figure 1.2 shows a generic block diagram of a digital-signature based video authentication system. The perceptual hash function h generates a hash $h(V, K)$ of the input video V by applying a secret key K. For additional security, $h(V, K)$ is encrypted with a *private key* K_{pr} in the PKCS and a digital signature of V is derived. For the use during the signature verification, the secret key K and other necessary information for hash computation may be also encrypted along with $h(V, K)$. The signature is sent to the user through a secured channel. On receiving the signature, the user decrypts it with an appropriate *public key* K_{pu} and extracts the original hash $h(V, K)$. The hash $h(V', K)$ of the received video V' is computed and compared with $h(V, K)$ by means of a distance measure $d(h(V, K), h(V', K))$. The authenticity of V' is decided by using a suitable threshold T according to:

$$\begin{aligned} d\left(h\left(V,K\right),h\left(V',K\right)\right) \leq T &\quad : \quad V' \text{ authentic, and} \\ d\left(h\left(V,K\right),h\left(V',K\right)\right) > T &\quad : \quad V' \text{ inauthentic.} \end{aligned} \tag{1.4}$$

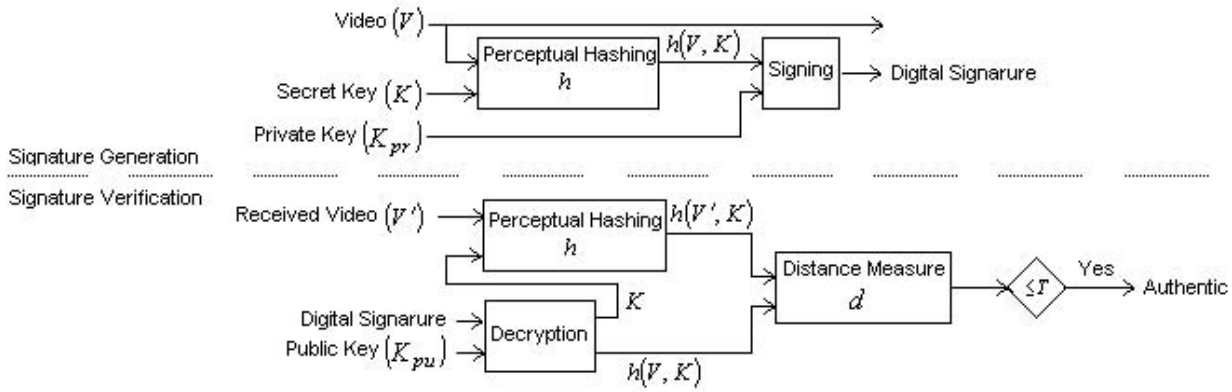


Fig. 1.2: Block-diagram representation of the signature generation and verification steps in a digital-signature based video authentication system

1.3 Perceptual Hashing of Scalably-Coded Video

Due to the recent developments in the field of *scalable video coding* (SVC) [19] and the convergence of various data networks, the robustness of a perceptual hash function against the scalability features of an SVC scheme becomes an issue. In that case, η_{cp} in Property 4 has also to include the distortions

due to the scaling of a scalable video bit-steam. This thesis focuses on the perceptual hashing for scalably-coded video.

1.3.1 Scalable Video Coding

SVC is based on the multi-resolution representation of digital videos. Given a video, a scalable coder delivers a *scalable* or *progressive* bit-stream consisting of layers that allow decoding at various levels of temporal, spatial and bit-rate or signal-to-noise ratio (SNR) resolutions [19], [20], [21], [22]. Thus, the SVC schemes provide temporal (resolution) scalability, spatial (resolution) scalability and bit-rate / SNR (resolution) scalability. The *base layer* of the bit-stream is independently decodable with reduced resolutions and the *enhancement layers* offer increasingly better qualities to the decoded base layer. Therefore, a scalable bit-stream can automatically adapt to the channel and terminal limitations. SVC is particularly useful in video transmission services with heterogenous clients and transmission scenarios with unpredictable throughput variations and/or substantial packet losses, surveillance applications for viewing and storing of video with limited resources, etc [21].

In practice, the heterogenous clients and transmission scenarios in the video transmission services are handled by using *transcoder*s. Video transcoders ensure interoperability between networks and systems by performing spatio-temporal and bit-rate resolution reductions [20]. Transcoders fall under two categories: pixel-domain and compressed-domain [23], depending on whether they operate on a raw video or on a compressed video. A transcoder operating in the pixel-domain fully decodes an incoming compressed bit-stream and re-encodes after reducing resolutions in the pixel domain. The complexity and the delay are its inherent drawbacks. Operating on partially decoded bit-streams, transcoders in the compressed-domain offer more efficient solutions for the real-time applications. For example, two popular video coding standards are MPEG-1 and MPEG-2 standardised by the Moving Picture Experts Group (MPEG). For an MPEG-1/2 video, the resolution-reduction operations can be performed after partially decoding the coded bit-stream. The reduction in the temporal resolution or the frame-rate can be achieved by directly dropping B- or P- frames [23]. Similarly, the spatial-resolution reduction or the frame resizing can be performed in the discrete cosine transform (DCT) domain by working on the 8×8 DCT blocks [24]. For bit-rate resolution reduction, the DCT coefficients can be quantized with a larger quantization step-size or some of the high-frequency DCT coefficients may be dropped [25].

The basic difference between SVC and the transcoding lies in their approaches in handling the same network and system adaptability issues. A scalable coder does not consider the transmission

requirements when specifying the data format during the encoding process. On the other hand, adapting a bit-stream according to the network and system constraints, the transcoders in the data delivery path maximise the end-user experience and the quality of service. With SVC, the resolution-reduction operations do not require decoding / coding of the coded bit-stream and consist of simple parsing operations on the bit-stream [22].

The SVC systems are either based on hybrid schemes or on spatio-temporal wavelet technologies [22]. The MPEG-4 [26] and H.264/advance video coding (H.264/AVC) [27] are examples of hybrid coding schemes. The MPEG, International Organization for Standardization (ISO) / International Electrotechnical Commission (IEC) and International Telecommunication Union - Telecommunication standardization sector (ITU-T) formed a Joint Video Team (JVT) to develop an SVC amendment for the H.264/AVC standard [27], [28], [29], [30]. Although the JVT has adopted an SVC model derived from the H.264/AVC technologies [29], [30], it has also set references for continuing experiments on the wavelet technologies for developing *wavelet-based SVC* (WSVC) systems [22].

The *3D discrete wavelet transform* (3D-DWT) may be an alternative means of achieving scalable coding. Video coding based on the 3D-DWT is getting much attention in recent times. Due to the intrinsic multi-resolution property of wavelet representation, the 3D-DWT based coding schemes can provide good temporal- and spatial- resolution scalabilities [31]. A WSVC scheme can accommodate the temporal and spatial scalabilities by effective exploitation of the multi-resolution property of the 3D-DWT. Although it has not yet been possible to establish the wavelet video coding as an alternative video coding standard for SVC, its performance appears on the rise when compared to previous attempts to establish credibly competitive video-coding solutions with respect to the hybrid coding approaches [22].

A hierarchically-structured bit-stream based on the multi-resolution structure can facilitate fast retrieval of a desired video from a database [32]. This thesis particularly considers developing video hash functions in the WSVC framework.

1.3.2 Decomposition of Video Using the 3D Discrete Wavelet Transform

The DWT projects a signal on a set of multi-resolution subspaces allowing a critically sampled representation of the signal in the transform domain and guaranteeing perfect reconstruction synthesis. In video coding using the 3D-DWT, a video is first divided into groups-of-frames (GOFs), usually each with 4, 8, 16 or 32 frames [33], [34]. Each GOF is considered independently. Depending on the order in which the temporal and spatial wavelet transforms are applied, there are two types of

wavelet video coding. In the 't+2D' approach, the frames in a GOF are temporally decomposed first, followed by the 2D decomposition of the temporal wavelet bands. In the '2D+t' approach, the order in which the transforms are applied is reversed: the 2D decomposition of the frames is followed by the temporal decomposition of the spatial wavelet bands. The MPEG has explored the 3D-DWT based video coding and has accepted the t+2D approach as the first working draft [35].

Consider a GOF G of size $N_1 \times N_2 \times N_3$, where $N_1 \times N_2$ is the frame dimension and N_3 is the number of frames in the GOF. Single level temporal decomposition of G results in one low-pass band $t1L$ and one high-pass band $t1H$, each comprising $\frac{N_3}{2}$ frames. The low-pass band is recursively decomposed to achieve decomposition at multiple levels. Let G be temporally decomposed up to the level u. This derives one temporal low-pass band tuL at the highest level of decomposition and the temporal high pass bands $tuH, t(u-1)H, \cdots, t1H$, one at each level of decomposition. After the temporal decomposition, the frames in each temporal band are decomposed spatially. At the first level of spatial decomposition, the temporal low-pass band tuL results in one spatio-temporal low-pass band $tuL - s1LL$ and three orientation-selective high-pass bands $tuL - s1LH$, $tuL - s1HL$ and $tuL - s1HH$, each of spatial size $\frac{N_1}{2} \times \frac{N_2}{2}$. At the second level of decomposition, spatio-temporal low-pass band $tuL - s1LL$ derives one low-pass $tuL - s2LL$ and three high-pass bands $tuL - s2LH$, $tuL - s2HL$ and $tuL - s2HH$, each of size $\frac{N_1}{2^2} \times \frac{N_2}{2^2}$. Thus, a multi-resolution representation of G is obtained by decomposing it recursively. The multi-resolution structure offers the temporal and spatial scalabilities. Figure 1.3 shows a GOF with 16 frames at the first levels of the temporal and spatial decomposition.

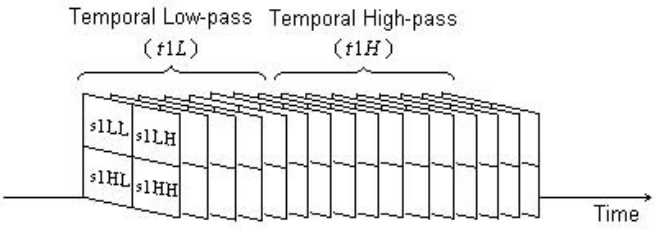


Fig. 1.3: The wavelet bands at the first levels of the temporal and spatial decomposition of a GOF with 16 frames using the 3D-DWT

The coefficients of discrete-wavelet decomposition at different levels can be computed using invertible filter banks. We consider the Haar wavelet functions for illustration in the following for simplicity. Suppose $f_k^{t(u-1)L}(n_1, n_2)$ and $f_k^{t(u-1)H}(n_1, n_2)$, where $n_1 = 1, 2, \cdots, N_1$ and $n_2 = 1, 2, \cdots, N_2$, respec-

tively represent the wavelet coefficients at the location (n_1, n_2) of the k^{th} frames in the temporal low-pass and high-pass bands $t(u-1)L$ and $t(u-1)H$. Each of the $t(u-1)L$ and $t(u-1)H$ bands contains $\frac{N_3}{2^{u-1}}$ frames. Further decomposition of the $t(u-1)L$ band derives the frames in the tuL and tuH bands according to the following two equations [36] for $k = 1, 2, \cdots, \frac{N_3}{2^u}$.

$$f_k^{tuL}(n_1, n_2) = \frac{1}{2}\Big(f_{2k-1}^{t(u-1)L}(n_1, n_2) + f_{2k}^{t(u-1)L}(n_1, n_2)\Big) \tag{1.5}$$

$$f_k^{tuH}(n_1, n_2) = \frac{1}{2}\Big(f_{2k-1}^{t(u-1)L}(n_1, n_2) - f_{2k}^{t(u-1)L}(n_1, n_2)\Big) \tag{1.6}$$

Again, the separable spatial decomposition of the k^{th} frame f_k^{tuL} can be modelled with the following six equations [36] for $n_1 = 1, 2, \cdots, \frac{N_1}{2^v}$ and $n_2 = 1, 2, \cdots, \frac{N_2}{2^v}$.

$$f_k^{tuL-svL}(n_1, n_2) = \frac{1}{2}\Big(f_k^{tuL-s(v-1)LL}(2n_1 - 1, n_2) + f_k^{tuL-s(v-1)LL}(2n_1, n_2)\Big) \tag{1.7}$$

$$f_k^{tuL-svH}(n_1, n_2) = \frac{1}{2}\Big(f_k^{tuL-s(v-1)LL}(2n_1 - 1, n_2) - f_k^{tuL-s(v-1)LL}(2n_1, n_2)\Big) \tag{1.8}$$

$$f_k^{tuL-svLL}(n_1, n_2) = \frac{1}{2}\Big(f_k^{tuL-svL}(n_1, 2n_2 - 1) + f_k^{tuL-svL}(n_1, 2n_2)\Big) \tag{1.9}$$

$$f_k^{tuL-svLH}(n_1, n_2) = \frac{1}{2}\Big(f_k^{tuL-svL}(n_1, 2n_2 - 1) - f_k^{tuL-svL}(n_1, 2n_2)\Big) \tag{1.10}$$

$$f_k^{tuL-svHL}(n_1, n_2) = \frac{1}{2}\Big(f_k^{tuL-svH}(n_1, 2n_2 - 1) + f_k^{tuL-svH}(n_1, 2n_2)\Big) \tag{1.11}$$

$$f_k^{tuL-svHH}(n_1, n_2) = \frac{1}{2}\Big(f_k^{tuL-svH}(n_1, 2n_2 - 1) - f_k^{tuL-svH}(n_1, 2n_2)\Big) \tag{1.12}$$

The notations $f_k^{tuL-svLL}$, $f_k^{tuL-svLH}$, $f_k^{tuL-svHL}$ and $f_k^{tuL-svHH}$ respectively represent the spatial low-low (LL), low-high (LH), high-low (HL) and high-high (HH) bands for the frame f_k^{tuL} at the spatial decomposition level v. The equations (1.7) and (1.8) operate on each column and divide a frame into two halves horizontally. The upper and lower halves are respectively the low-pass and high-pass bands along the vertical direction. When operated with (1.9) - (1.12), these two halves obtain one low-pass band $f_k^{tuL-svLL}$ and three high-pass bands $f_k^{tuL-svLH}$, $f_k^{tuL-svHL}$ and $f_k^{tuL-svHH}$.

The decomposition process is realisable using digital *finite impulse response* (FIR) filters. From (1.5), the frames in the tuL band can be obtained by passing the frames in the $t(u-1)L$ band through a FIR filter operating along the temporal direction with the impulse response [36]

$$\widetilde{h}(k) = \frac{1}{2}(\delta(k) + \delta(k-1)) \tag{1.13}$$

and by retaining the alternate frames at the output (*decimation*). Similarly, from (1.6), one can obtain the frames in the tuH band by filtering the frames in the $t(u-1)L$ band by using a FIR filter with impulse response [36]

$$\widetilde{g}(k) = \frac{1}{2}(\delta(k) - \delta(k-1)) \tag{1.14}$$

and followed by the decimation. The filters $\widetilde{H}$ and $\widetilde{G}$ are called *analysis* or *decomposition filters*.

The spatial decomposition of each frame in the temporal bands described by (1.7) - (1.12) can be achieved by applying the same set of filters in (1.13) and (1.14) in the direction of n_1 followed by the decimation and then in the direction of n_2 followed by the decimation. As an example, the bands at the first levels of tempoal and spatial decomposition of a GOF can be obtained using the filter bank in Figure 1.4. The references [36] and [37] are good texts on the wavelet theory.

1.3.3 Wavelet-based Scalable Coding

Researchers are trying to exploit the intrinsic multi-resolution structure of the wavelet transform for SVC. The wavelet-based image compression techniques, namely, the embedded zerotree wavelet (EZW) [38], Set Partitioning in Hierarchical Trees (SPIHT) [39], embedded zero block coding (EZBC) [40], embedded morphological dilation coding (EMDC) [41], [42], etc., are efficient in terms of the bit-rate scalability vis-à-vis the computational complexity [22]. All these schemes use the zero-tree hypothesis. In the JPEG 2000 standard [43], the embedded block coding with optimized truncation (EBCOT) algorithm [44] has been adopted. JPEG 2000 provides good scalability and high coding efficiency [22].

The temporal extensions of SPIHT [33], EZBC [45], [46], EMDC [41], [42], [47], EBCOT [48] are used in the WSVC systems [22]. Among others, the spatio-temporal subband decomposition is exploited in designing a low bit-rate video coding scheme in [49]. A wavelet-based spatially scalable coder, which is also robust over a wide range of bit-rate, is presented in [50]. The video codec

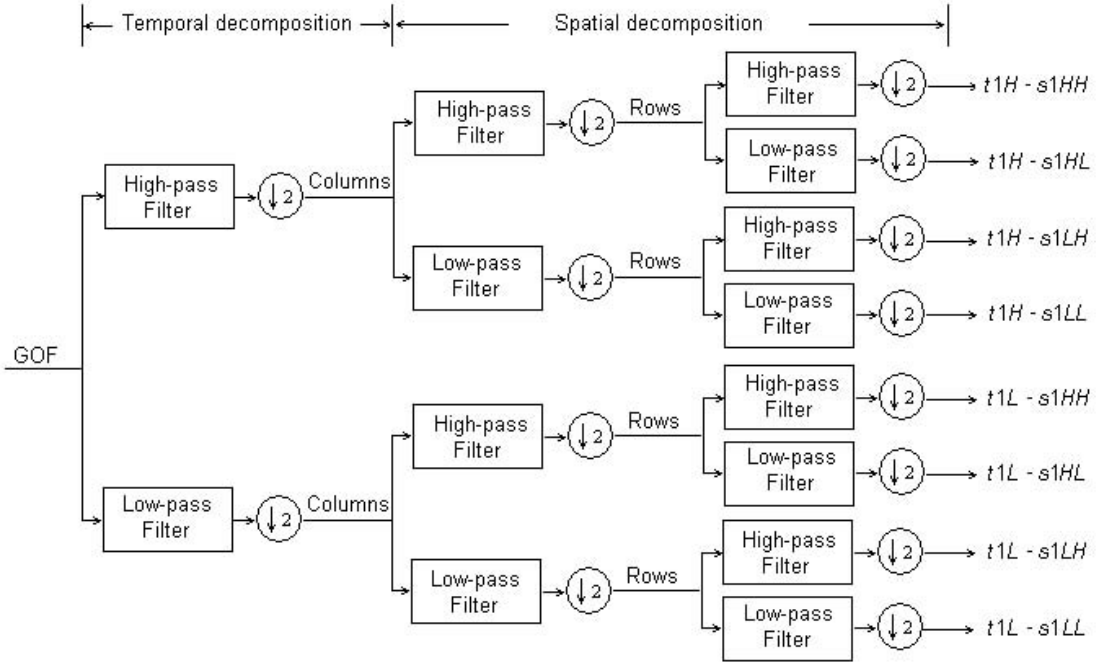


Fig. 1.4: The filter-bank realisation of the one level of temporal and one level of spatial decomposition of a GOF by using the 3D-DWT

in [51] based on the subband decomposition provides a combination of temporal, spatial and bit-rate scalability. Another fully-scalable SPIHT-based video coder is presented in [52]. A near lossless wavelet-based video coder providing the bit-rate scalability is presented in [53]. The reference [32] is a good document on the wavelet-based image and video coding.

The intrinsic multi-resolution structure of the 3D-DWT may be exploited for hashing of video in the wavelet domain. The temporal- and spatial- resolution reductions in the case of a wavelet-coded GOF can be performed in the powers of two by selectively dropping temporal and spatial high-pass bands respectively. As mentioned above, this multi-resolution structure is exploited in many wavelet-based video coders. Hence, for the robustness of a perceptual hash function against the scalability features of the 3D-DWT or the WSVC schemes, the hash function may be designed to compute video hashes from the spatio-temporal low-frequency contents of videos. In that case, the hash function will also be robust against the transcoding operations in the 3D-DWT domain.

There are very few perceptual hash functions in literature which are robust against the scalability features of an scalable coding system and against the compressed-domain transcoding operations. The image hash function in [54] is compatible to the scalable coding framework and is robust against the scalability features of the JPEG 2000 coding standard. The video hashing in the 3D-DWT

domain with robustness against the scalability features of WSVC is an unexplored area. The video hashing algorithm in [55] is robust against the transcoding operations performed in the DCT domain. Although the 3D-DCT based hash functions in [56] are robust against the transcoding, they require full decompression of a coded bit-stream for hash computation thus increasing the computational complexity.

1.4 Motivation and Problem Definition

It is observed in the previous section that the 3D-DWT has potentials for the WSVC applications / standards. Existing perceptual hash functions for video do not address the robustness of the hash against the scalability features of the WSVC schemes. It is, therefore, worthwhile to develop hash functions for video compressed in the WSVC framework.

The thesis addresses the following two problems:

i. Different spatio-temporal bands of the 3D-DWT decomposition of a video represent the video at different resolutions. One or more of these bands may be possibly used to represent the perceptual content of the video. The thesis examines the representation of the content of a video at the GOF level by the spatio-temporal bands derived from the 3D-DWT decomposition of the GOFs.

ii. Once the problem of representation is addressed, the next issue is to design perceptual hash functions from the perceptually-representative spatio-temporal band. These hash functions should satisfy the different requirements of a perceptual hash function with the additional requirement of the robustness against the scalability features of the 3D-DWT based scalable coding.

1.5 Outlines of the Thesis

The organisation of the rest of the thesis is as follows:

In Chapter 2, a comprehensive review of the previous works on the perceptual hash functions for image and video is presented. It discusses about the un-addressed research issues in video hashing and presents the motivation of this work.

Chapter 3 explores the possibility of using temporal and spatio-temporal bands of the 3D-DWT decomposition of a video for representing the content of the video. Detailed experimental results

are presented to demonstrate the effectiveness of using the spatio-temporal low-pass bands for the representation of video content.

Chapter 4 extracts a workable hash from a spatio-temporal low-pass band of the 3D-DWT decomposition of a video. Detailed experimental results are presented to demonstrate the performance of the proposed perceptual hash function including the robustness against the scalability features of the 3D-DWT based scalable coding.

Chapter 5 focuses on the compactness of the perceptual hash and presents a perceptual hash function in the 3D-DWT domain with good diffusion and confusion properties. Experiments are performed to examine the desired properties of the hash function. A detail analysis of the performance is presented.

Chapter 6 summarises the contributions of the thesis and suggests the scopes for future investigation.

Chapter 2

Perceptual Hashing: Current Practices and Issues

The cryptographic hashing of text messages is a matured field. It is observed in Chapter 1 that MD5 and SHA-1 are exemplary hash functions for very secured hashing of text messages. Although many fundamental questions remain open [5] [6], there are many publications on image hashing. Video hashing is comparatively a new area of research [57].

Many of the reported perceptual hash algorithms for video apply perceptual hash functions for image on each frame of a video. The frame hashes are concatenated to a hash of the video. One serious drawback of the frame-by-frame hash computation is that the temporal subsampling of the video affects the hash severely. Some perceptual hash functions represent the video with *key frames* [58] selected from the frames in the video and extract a hash of each key frame by applying image hashing. The hashes of the key frames are collectively considered as a hash of the video. In this case also, temporal scaling of the video may severely affect the hash. Others treat the video as a single entity and summarise the content of the video into *representative frames* [57] by taking into account the spatio-temporal content of the video at a time. A hash of the video is derived from the hashes representative frames.

This chapter classifies the perceptual hash functions into two broad categories based on their domain of working: *hash functions in the pixel domain* and *hash functions in the transform domain*. It presents a review of some of the perceptual hash functions for image and video available in the literature. As a large-scale experimentation is required to evaluate the relative performances of the hashing algorithms, the conclusions made here are based on the results reported in the literature.

2.1 Classification of the Perceptual Hash Functions

Depending on the hash computation strategies, many authors [3], [5], [6], [7] divide the perceptual hash functions for images into four broad categories. The four categories are as follows:

i. Perceptual hash functions based on the pixel statistics: In this category, a hash of an image is computed from the pixel statistics of the image. The pixel statistics are usually obtained from the image histograms and/or various moments.

ii. Perceptual hash functions based on the low-level information: The low-level features such as edges, corners, etc. are visually salient and can describe the content of the image. The perceptual hash functions in this category extract the hash from these features.

iii. Perceptual hash functions based on the relation: In the third category, the hash is derived based on some relation among the image features. A number of hash functions exploit the relation among the histogram bins or the transform coefficients in the DCT and DWT domain.

iv. Perceptual hash functions based on the low-pass content: It is not possible to change the perceptual content of the image without affecting the low-pass content. The low-pass information in the image is used by many hash functions for hash computation.

This classification can also be extended to the perceptual hash functions for video. The perceptual hash functions generate hashes either from the raw images / videos or from the transform coefficients of the images / videos. In the present study, the classification of the perceptual hash functions based on their domain of working will be useful. Hence, we divide the perceptual hash functions into two broad categories:

i. Perceptual hash functions in the pixel domain: A perceptual hash function in the pixel domain computes a hash of an image or a video from the pixel characteristic of the image or video. The pixel characteristic may be represented with the histogram, moments, edges, corner or shape information, etc. These information describe the content of the image or video.

ii. Perceptual hash functions in the transform domain: In this category, hashes are extracted from the transform coefficients of the image or video. Many of these hash functions exploit the low-pass content of the image or video for hash computation. Some of them exploit the invariant relationship among the transform coefficients. The statistics of the transform coefficients are also used to compute hashes.

The compression has become almost a standard feature in the multimedia systems. The image / video compression coders apply transformations like DCT (e.g. in the JPEG coder for image and MPEG-x coders for video) and DWT (e.g. in the JPEG 2000 coder for image). A perceptual hash function in the transform domain may be advantageous over the ones in the pixel domain if it does not require full decoding of a coded image or video bit-stream during the hash computation. It is implicit that the processing and time complexities reduce significantly if the full decoding is not necessary. These complexities are critical when the hash function is used in the real-time applications.

The generic block-diagrams of the two types of hashing for video are shown in Figure 2.1. In the pixel-domain category shown in part (a), the content-based features like histograms, moments, edges, corners, shape, etc. of a video are extracted. A hash of the video is then computed from these features. In part (b), the hash function applies a transformation on the video and extract features in the transform domain. The low-pass transform coefficients are often selected as features. Finally, the hash is computed from the transform-domain features.

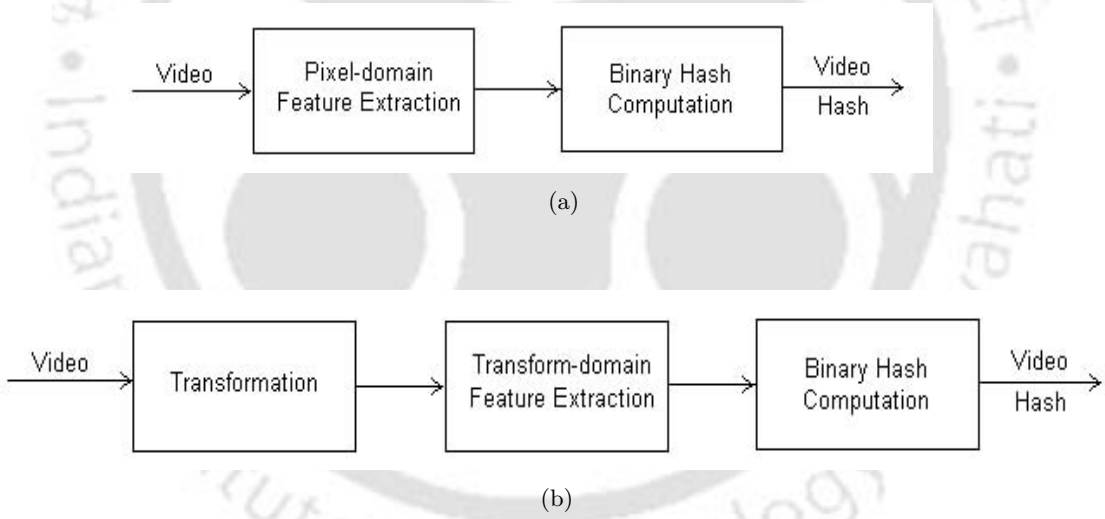


Fig. 2.1: The generic block-diagrams for video hashing in the (a) pixel domain (b) transform domain

2.2 Perceptual Hashing of Images: A Review

There are many algorithms available in literature for perceptual hashing of images. A brief review of some of the algorithms is presented in this section.

2.2.1 Hash Functions in the Pixel Domain

Schneider and Chang [59] divide an image into blocks of variable sizes and extract hash bits using the statistics of the intensity histograms of the blocks. Although the hash shows very strong diffusion property, its poor performance against the JPEG compression may not be acceptable. To enhance the robustness, the image may be represented with the mean of the intensity histograms. But, the increased robustness can be achieved only at the cost of sensitivity to malicious modification. The general drawback of the histogram-based hash functions is that an image can be modified without changing its histogram [6]. For better hash security, the blocks may be considered overlapping.

The complexity of the hash comparison reduces when the moments are used as features. In [60], Alghoniemy and Tewfik use the geometric and invariant moments to provide hashing solutions for image. As a result, the hash is invariant under the translation, scaling, reflection and orientation of the image. It is also reported to be robust against the filtering, compression, noise, etc.

Dittman et al. [61] present an edge-based hash function which determines the edge characteristic of the image (or single video frame in the case of video) with the Canny edge-detector and transform the characteristic into a binary edge pattern. The authors use the variable length coding (VLC) for data reduction while producing a feature code for the image. Although the approach has good diffusion property, it does not work properly under high compression, quantization and scaling as the edges slightly move from the original locations.

Xiang et al. [62] propose a hashing solution where the histogram bins are used to derive a hash of an image. It is observed that the relationship in the number of pixels among the groups of two different bins of histogram shape is invariant. This invariance is exploited to compute a hash of the image. Experimental observations suggest satisfactory performance of the hash against various geometric deformations.

2.2.2 Hash Functions in the Transform Domain

Venkatesan et al. [63] compute an image hash from the statistics of the transform coefficients in the wavelet-decomposed image. Each wavelet band is randomly tiled and the hash bits are obtained by binarising the average values of the tiles in the coarse subband and the variances of the tiles in the other subbands. The hash function is highly robust against the common image-processing distortions. But, it cannot detect all malicious modifications [6].

In another approach, Roover et al. [58] radially project the pixels in an image on concentric lines passing through the center of the image. A normalised and zero-mean vector is formed with the

variances of the pixel luminances in each radial line. For decorrelation of the vector elements, the DCT is applied on the vector. To reduce the hash length, the first 40 DCT coefficients are passed through a 8-bit quantizer deriving a 320-bit hash. This radial hashing (RASH) algorithm is reported to be robust against the geometrical distortions like frame rotation and scaling. Robustness is also reported against the low-pass filtering and lossy-compression operations.

In the hash function proposed by Bhattacherjee and Kutter [64], visually salient image features are extracted by using the *scale interaction model* with the Mexican-Hat wavelets. The authors compute the differences between the wavelet coefficients at two different scales. The local maxima of the differences correspond to a set of potential feature points. A point of local maximum is retained as a feature point if the variance of the image pixels in the neighbourhood of the point is higher than a threshold. The advantages of this hash function are the compact nature of the hash and the robustness against the geometric scaling in powers of two. It is reported that the hash function is compression tolerant and has good diffusion property. However, the robustness against the lossy compression is unclear [66].

Monga and Ivan [3], [4] propose an iterative feature extractor to extract significant geometry-preserving feature points for an image. The features are extracted by applying an *end-stopped wavelet* on the image in different orientations. The end-stopped wavelet kernels capture the essential and robust attributes of human perception. For introducing randomness and enhancing robustness against the perceptually insignificant perturbations, feature extraction is followed by a probabilistic quantization. Due to the intrinsic sensitivity of the feature detector, the hashing algorithm shows very good diffusion property.

Chang et al. [65] find salient points of an image by using the *3D scale-space representation*. The original image is then highly compressed under a weighted norm determined by a weighing function constructed from the salient points. The compressed image is used as a hash of the image. By using the weighing function, the hash function recognises that the important content information is not uniformly distributed across the image. This is helpful because the illegal operations on images are usually localised and the permissible operations are global in nature.

Lin and Chang in [66], [67] exploit the invariant relationship between two corresponding coefficients in a pair of DCT blocks of an image to derive a hash of the image. These relationships are mathematically shown to be robust against the JPEG compression regardless of the compression ratio and the number of recompression cycles. The hash function can sense the malicious activities on images successfully. However, it is not clearly mentioned whether it can survive under the

content-preserving operations other than the JPEG compression. In [68], the authors extend the above concept to handle the adaptive compression-rate control using the variable quantization tables in the later JPEG or MPEG coding standards.

In [69], Lu and Liao present a structural hashing solution for image by identifying the stable relationships between the parent-child pairs of the transform coefficients of an wavelet-decomposed image. The authors observe that the difference in the magnitudes of the coefficients in a parent-child pair is robust against many content-preserving operations. It is reported that the hash function performs satisfactorily against the content-preserving operations and has good diffusion property as well.

The perceptual content of a block in an image cannot be changed without affecting the low-frequency DCT coefficients of the block. In [70] and [71], Fridrich et al. divide the image into 64×64 blocks. Each image block is projected onto 50 numbers of low-frequency and DC-free random smooth patterns generated by using a secret key. The absolute values of the projections are compared with a suitably chosen threshold such that the number of zeros and ones in the hash are approximately equal. The hash function shows fair robustness against the content-preserving operations and detects the malicious modifications in a good way.

In [72], Mihcak and Venkatesan present a hash function for image that uses an iterative filtering approach to binarise the lowest resolution coefficients of an wavelet-decomposed image. The iterative filtering minimises the presence of 'geometrically weak components' and enhances the 'geometrically strong components'. The authors observe that the significant geometric features of an image are preserved by the hash function under small perturbations to the image.

In an another approach, Swaminathan et al. [73] apply the histogram equalisation on the low-pass filtered and downsampled version of an image. The resulting image is Fourier transformed and represented in the polar coordinate system. This is followed by linear additions of the equidistant points on the angle axis, where the weights are randomly generated using a key. The resultant vector is quantised and gray coded to derive the hash bits. This hash function is inherently rotation invariant as the hash is computed considering the transform coefficients that are rotation invariant.

Uehara et al. [74] present a JPEG tolerant hash function, where the level of protection can be adjusted so that higher security can be achieved at the cost of the length of the *message authentication code* (MAC) [1]. The 8×8 blocks of an image is divided into groups using a secret key. For enhanced security, the groups may be considered 'linked', i.e., each block may be contained in more than one group. The MAC of the image consists of features that are obtained by encoding a linear

combination of the DCT coefficients at each frequency in the blocks of each group. The features may not be computed at all the frequencies as an image with few details does not need high frequencies to be protected. During the binarisation of the features, the length of the MAC is chosen depending on the acceptable quality level of the JPEG compression for which the hash verification should produce correct result. As the authors mention, one drawback of the hash function is that it may not tolerate acceptable manipulations other than the JPEG compression.

The hash function, proposed by Ahmed et al. [75], extracts a hash of an image by using the wavelet coefficients in the LL, LH and HL bands of the wavelet-decomposed image. The authors divide the image into non-overlapping square blocks and apply random 'intensity transformation' by modulating the pixels with an integer generated using a secret key. Applying the DWT, the intensity-transformed blocks are decomposed fully. For each block, a hash is derived by adding separately the LL coefficient with the LH and HL coefficients. An intermediate hash is formed by putting together the hashes of the blocks and the same is permuted with another secret key for security. The hash function is reported to be resilient against the JPEG compression and high-pass filtering, and sensitive to the malicious modifications. Although the hash function can detect the malicious modifications very well, it is not robust enough against the non-malicious modifications.

In [54], Sun et al. present a hashing strategy for verifying the authenticity of JPEG 2000 images quantitatively and securely in terms of the 'lowest authentication bit-rate' (LABR) which should be always smaller than the 'compression bit-rate' (CBR). Given a target 'compression bit-rate' (CBR), the EBCOT block in the JPEG 2000 coder provides exact information about the fractionalised bit-planes of data to be included in the final bit-stream. Two measures are selected as invariant features: the states of the passes or the fractionalised bit-planes of the most significant bits (MSBs) and the estimated distortion associated with each pass. The features are coded using error correction coding (ECC) for attaining robustness against the content-preserving manipulations. The EBCOT block being the last processing unit in the JPEG 2000 coder, the image can be authenticated even after truncating or parsing the compressed bit-stream. The hash function is reported to be robust against recompression at different bit-rates, limited transcoding operations, etc.

2.3 Perceptual Hashing of Video: A Review

In comparison to the image hashing, only a few perceptual hash functions are reported for hashing of video. Although the perceptual hash functions for image may be applied frame-by-frame for hashing

of video [59], [61], [64], the large sizes of the hashes make it impractical. Moreover, frame-by-frame hashing is computationally expensive and the subsampling along the temporal axis strongly affects the hashes. Often, hash functions for image are applied to the selected key frames of a video [58]. For simplicity, many authors choose the first, the middle or the last frame in a *video shot* as a key frame of the shot [76]. Others choose the key frames on the basis of a minimum difference criterion on consecutive frames [77].

The frame and key-frame based video hashing approaches ignore the temporal information contained in the video. By nature, video data have high correlation in the temporal direction. The temporal correlation in a video may be exploited while computing a hash of the video. In a few hashing solutions [8], [56], [57], the content of the video is summarised into representative frames by considering the spatio-temporal content at a time. In the following, a comprehensive review of the perceptual hash functions for video is presented.

2.3.1 Hash Functions in the Pixel Domain

In [78], Radhakrishnan and Bauer propose a hash function based on the moving regions active in each pair of the adjacent video frames. A hash of a video is obtained by considering together the hash bits generated from the pairs of frames. The hash generation steps in the following take a pair of frames as input and output a hash of the pair.

- Compute the absolute difference of the frames capturing the temporal variations in them.
- Downsample and crop the absolute difference image for invariance against the addition of graphics / boxes on the corners of the frames.
- Tile the cropped image and compute the averages of the blocks generating a 'feature matrix'.
- Using a key, generate 36 zero-mean uniformly-distributed random matrices with their elements in the range [0 1]. The dimension of the random matrices is the dimension of the feature matrix.
- Project the feature matrix on to the random matrices.
- Extract hash bits from the projections using their median as the threshold.

The authors observe that the hash survives MPEG compression, colour-space conversions, intensity variations, noise additions, small rotations, etc. Dissimilar videos are also discriminated very well.

In an another approach, Atrey et al. [79] extract key frames of a video based on the differential energy between the frames. A hash of the video is computed from the key frames. The hashing strategy is scalable to three hierarchical levels: key-frame, shot and video. Following are the steps for computing the hash.

- Segment the input video into shots.
- For each shot, consider the first frame as one key frame. Identify other key frames based on the computation of the differential-energy of the pixel luminance values.
- Quantize the luminance values of the pixels of all frames and retain only the unique quantized values ignoring the repeated ones except for the key frames.
- Compute an interpolating polynomial by using the secret sharing from the non-key frames between each pair of key frames. Construct a new polynomial by replacing the coefficients of the interpolating polynomial with their crypto-hash value. Extract a secret frame at the key-frame level by extrapolating the new polynomial at a position decided by a secret key.
- The extrapolated secret frames and the key frames in each shot generate a secret frame at the shot level following the previous step.
- The secret frames at the shot level derives a master secret frame at the video level.

This hashing strategy fits to the streaming video scenario and can be used for video identification. Results are also cited to demonstrate its use in detecting face tempering.

Mucedero, Lancini and Mapelli [14], [15] propose a hashing algorithm for identification of videos in databases. The algorithm pre-processes a video and computes a hash at the frame level by extracting robust features. The following are the steps for computing a hash of a frame.

- Extract the luminance component of the frame and re-sample it to the size 288×352.
- Apply a low-pass filter and downsample the frame to the size 144×176.
- Construct a 'variance matrix' by computing the variance of the pixels in a 15×15 block around each pixel in the preprocessed frame.
- Split the variance matrix into 16×16 blocks. For each block, extract the location of the minimum variance. In case the minimum appears for more than once in a block, count the

number of appearances and consider the median location of the locations of the minima. If the minimum appears in a block boundary, reject the block.

- The count and the locations of the minimum variances constitute the hash.

The hashing algorithm is reported to be efficient at different bit rates.

A video hash function based on the 'centroid of gradient orientations' is proposed by Lee and Yoo in [80]. The authors observe that the centroids of gradient orientations are pairwise independent and robust against many common content-preserving operations. The hash algorithm can be summarised as follows.

- Resample a given video in the time direction and convert it to a gray-scale one. Normalise the frames to a fixed size.
- Divide each resized frame into blocks.
- For each block in each frame, compute the gradient vector at every point and find the centroid of the gradient orientations.
- Construct a feature vector by considering together the centroids obtained in the preceding step.

As the gradients are based on pixel differences, the hash function is inherently robust against the global changes in the brightness, colour and contrast. It is reported to be robust against the frame-rate change, lossy compression, low-pass filtering, noise, etc. But, its performance degrades under geometric transformations, such as frame rotation and cropping.

Shivadas and Gauch in [81] use 27 colour moments to characterise the content of every frame in a video. The authors compute the features of a video frame as follows.

- Compute the mean, standard deviation and skew of the frame in each of the red, green and blue colour channels. This derives nine spatial moments.
- Compute the mean, standard deviation and skew of the frame along the two spatial directions individually in the red, green and blue colour channels. This derives 18 one-dimensional moments.
- Round the moments to the nearest integer in the range [0 255].

The authors claim that the representation is compact, easy to calculate and robust against a range operations on video. This moment set is also claimed to be less sensitive to noise.

2.3.2 Hash Functions in the Transform Domain

Oostveen et al. [8] extract features of a video from the luminance components of the video frames. A robust hash of a video is computed by applying the 2×2 spatio-temporal high-pass Haar filter according to the following steps.

- Divide each frame in the video into blocks of size 64×64 and compute the mean of the pixel luminances in each block. Arrange the averages in the raster-scan order.
- Compute the difference between each pair of consecutive averages to eliminate the effect of brightness modifications and obtain the differential spatial averages.
- Compute the differences between the corresponding (spatial) differential averages in each pair of consecutive frames in order to reduce the temporal correlation.
- Consider the signs of the spatio-temporal differential averages as the hash of the video.

The hash is used for identifying the video segments. The hash algorithm localises a segment in a movie with a low false detection rate. It is found to be robust against the content-preserving operations. The robustness performance of the hash may be enhanced by adopting a soft decision rule during the hash comparison, i.e., by making the similarity decision on the basis of the most reliable bits derived from the larger spatio-temporal differential averages. But, this is achieved at the cost of complicated searching and larger bandwidth requirement.

Roover et al. [58] extend the RASH algorithm discussed earlier for image hashing for the hashing of video. The authors represent a video with key frames extracted on the basis of the minimal local-disparity measurement. A hash of a key frame is computed according to the following RASH algorithm.

- Distribute the pixels in the key frame into 180 lines passing through the center of the frame with a separation of 1° in their angular orientations.
- Compute the variance of the pixel luminances on each line.
- Consider the 180 variances together, subtract their mean from each of them and divide with their standard deviation to derive a zero-mean and normalised radial variance (RAV) vector.
- Apply the DCT on the elements in the RAV vector. Consider the first 40 low-frequency DCT coefficients and quantize them using a 8-bit quantizer. This results in a 320-bit hash for the key frame.

It is reported that the hash is robust against the temporal subsampling, frame rotation and scaling, low-pass filtering and lossy compression.

Ahmed and Siyal [82] generate a hash of a video on a frame-by-frame basis using the intensity transformation described in [75]. Features of an intensity-transformed frame are extracted in the DCT domain. The hashing process is as follows.

- Divide the luminance plane of a frame into non-overlapping blocks.
- Modulate the pixels in each block with an unique integer derived by using a secret key.
- Perform DCT on the intensity-transformed blocks.
- Form each DCT block, obtain a scaler feature by adding the DC coefficient and the first AC coefficients from the first row and first column.
- Quantize the features by setting the quantization interval according to the maximum allowable change in the features due to non-malicious operations on the video.
- Concatenate the quantized values of the features and a 128 bit *salt* to derive the hash of the frame.

Note that the salt is added to enhance the fragility of the hash against the malicious activities. The authors demonstrate that the hash function is resilient against the codec variation and the low bit-rate encoding. It detects and localises spatial and temporal tempering.

The video hash function proposed in [83] by Uehara et al. is an extension of the image hash function in [74]. It can tolerate MPEG compression to a designated level. Each I-frame in a video is handled independently and features are extracted as follows.

- Divide the I-frame into 8×8 blocks and distribute into groups using a secret key.
- Perform 2D-DCT on each block.
- At each frequency, find the weighted sum of the coefficients of the blocks in each group.
- Encode the quantized weighted sums to generate binary strings, called the 'feature codes'. Decide the precision of binary representation at each frequency to determine the interval of the acceptable quality level for the MPEG compression.
- Obtain a MAC for the I-frame by putting the feature codes together.

If the P- and B- frames are considered independently similar to the I-frames, the size of the MAC for the video will be large. Noting that the P- and B- frames depend on the I-frames and it is sufficient to verify frames at a rate such that the visible changes become detectable, feature codes are computed for the every-other decoded P- and B- frames by the same procedure as the one used for the I-frames. To restrict the malicious modifications, the groups may be considered linked instead of disjoint. The length of the MAC for one frame is proportional to the number of groups. However, using of smaller number of groups reduces the hash security and hence, small malicious modification may remain undetected.

In [84], He et al. propose a hashing solution for MPEG-4 video, where a set of angular radial transformation (ART) coefficients are selected as the features of a video. The ART is a region-based shape descriptor and is invariant against scaling, rotation, translation and various types of shape distortions. It is also robust against the segmentation process. The steps for hash computation are:

- Retain a set of 15 ART coefficients per frame of a video that are of large magnitudes but do not suffer much under different scalings.
- Quantize 4 coefficients with large magnitudes into 5 levels and the other 11 coefficients into 3 levels to derive a binary feature vector.

The authors report that the hash is robust against the MPEG-4 compression and MPEG-4 object manipulations such as scaling, rotation and translation.

Sun et al. [55] present a configurable hash function for MPEG video which is robust against the video transcoding operations in the DCT domain. Based on the given transcoding operations and the 'authentication strength', invariant frame-based features from the DCT coefficients are extracted. During feature extraction, the authors use the invariant property of the DCT coefficients in the JPEG compression. If a DCT coefficient in an image is modified to an integral multiple of a quantization step size, which is larger than the step size used in later JPEG compression, the coefficient can be exactly reconstructed after the later JPEG compression. This property holds good for the I-frames of an MPEG video because the compression of an I-frame is exactly similar to the JPEG compression. The authors further observe that this property is also kept for the P- and B- frames in the video if the same predictive coefficients can be acquired during the coding and decoding of the video. To derive a hash resilient against the quantization-based transcoding, the following steps are to be followed.

- Decode the given MPEG-coded video and derive the 8×8 DCT blocks frame-by-frame.

- Form a feature set of the DC coefficients in the blocks of each frame.
- Quantize the feature set.
- Cryptohash the quantized feature set of the current frame and the hash of the previous frame.

To handle the frame-dropping during the transcoding, the hashes generated from the previous frames are embedded into the current frame using watermarking. For robustness against the frame-resizing based transcoding operation, feature extraction and watermarking are carried out in the *quarter common intermediate format* (QCIF) instead of the *common intermediate format* (CIF).

In [56], Coskun et al. present two perceptual hash functions in the 3D-DCT domain. While one hash function uses the classical basis set of the 3D-DCT, the other uses a set of random bases. In both the cases, a video is first converted into a 'standard' video signal of fixed spatio-temporal dimension $32 \times 32 \times 64$ via smoothing and subsampling. Temporal and spatial smoothing operations are performed by using 1D and a 2D Gaussian filters respectively. A hash of the video is computed from the standard video signal as follows.

- Apply the 3D-DCT to the standard video signal.
- Extract a $4 \times 4 \times 4$ block of transform coefficients in the low-frequency region. To enhance the uniqueness among the similar but not identical video sequences, exclude the lowest frequency coefficients in the three directions.
- Binarise the 64 DCT coefficients using their median as the threshold.

In the second approach, random bases are generated by using cosinusoidal signals with key-dependent random frequencies. Unlike the increasing frequency pattern observed in the 3D-DCT, the frequency pattern in this case is random. For robustness of the hash function, the random bases are low-pass filtered in all directions. The steps for hash generation are:

- Generate $32 \times 32 \times 64$ random bases using cosine signals with key-dependent random frequencies.
- Filter the bases using a 5-tap temporal and a 5×5 spatial low-pass filters.
- Project the standard video signal onto the random bases.
- Select 64 components obtained from the projections and binarise the components using their median as the threshold.

The authors report that both the hash functions exhibit robustness against the common signal processing operations on video as well as against the limited geometric distortions. The hash function using the classical basis set performs better than the hash function using the random bases.

Malekesmaeili et al. [57] extract a hash of a video from the *temporally informative representative image*s (TIRIs) containing both the temporal and spatial information in the video. For resistance against the scaling and frame-rate variations, the video is resampled in the temporal direction and resized spatially. The re-sampling and resizing operations are preceded by spatio-temporal low-pass filtering to avoid aliasing. The resulting video is segmented into shorter segments and the weighted averages of the luminance values of the pixels in each segment are computed in the temporal direction. This results in one TIRI per segment. Any image hashing algorithm can be applied on the TIRIs to extract a compact hash of the video. For experimentation, the authors apply the DCT-based hashing algorithm in [56] with minor modification for application on each TIRI. It is observed that the resulting hashes are robust against the Gaussian noise, frame dropping, brightness and contrast modifications, rotation, spatial shifts, etc.

2.4 Discussion

It is observed that a number of perceptual hash functions have been developed for hashing of image and video. While the perceptual hash functions in the pixel domain naturally require full decoding of coded image or video bit-streams, all the hash functions in the transform domain also cannot deal with partially decoded bit-streams. A few of the video hashing algorithms [55], [68], [83], [84] can handle partially decoded MPEG bit-streams.

As pointed out in the previous chapter, the 3D-DWT is becoming more and more popular in video compression due to its intrinsic spatio-temporal scalability. The above review shows that perceptual hashing of video in the 3D-DWT domain has not been properly addressed.

A perceptual hash function derived from a spatio-temporal low-pass band of the 3D-DWT decomposition of video may be naturally robust against the WSVC. As a special mention, the perceptual hash function in [54] is compatible to the scalable coding framework and is robust against the scalability features of the JPEG 2000 image coding standard. The video hash algorithm presented in [55] is robust against the transcoding operations in the DCT domain. With this background study, the next chapter investigates content-based representation of video by the spatio-temporal bands of 3D-DWT decomposition.

Chapter 3

Content-Based Representation of Video Using 3D Discrete Wavelet Transform

Perceptual hash functions for image are often extended to compute video hashes on a frame-by-frame basis [59], [61], [64]. Many perceptual hash functions for video also operate at the frame level [14], [15], [82]. In these approaches, a hash of a video is obtained by combining the hashes of the frames in the video. The advantage of a perceptual hash function operating at the frame level is that it can distinguish temporal contents in distinct video sequences at the frame level. But, the large hash size and the computational complexity are its two inherent drawbacks [58]. It is also sensitive to frame-dropping [56], [58]. To overcome these limitations, the independence of the shots in a video is exploited and each shot is represented by a key frame selected from the frames in the shot [58], [79]. The hash of the key frame is used as a hash of the shot. In another approach, representative frames for a video segment are extracted based on the spatio-temporal content of the segment [8], [56], [57]. A hash of these frames is used as a hash of the segment. The success of such hash functions broadly depends on the selection of the key frames or extraction of the representative frames.

Video hashing is one of the alternatives for authenticating video content or identifying video in a database [8]. As discussed in the previous chapter, in the video identification applications, one may wish to know about the origin of a query segment. The straightforward method for this is to compare the given segment with the segments of the videos in the database. The involved computational complexity and the large memory requirement are drawbacks of this approach. One way to overcome these drawbacks is to tag the video segments in the database with representative frames. The representative frames of the query segment is compared with those of each segment in

the database. But, the volume of bits representing each segment is still large for use in identification applications. A practical solution is to use perceptual hash function to compute a hash from the representative frames of a segment. The origin of the given segment is determined by comparing the hash of the query segment with the hash of each segment in the database [8], [14], [15].

This chapter considers to extract representative frames for video using the 3D-DWT. It first reviews in brief some of the available techniques for selecting key-frames of a video. A description of some of the existing methods for extracting representative frames is also presented. The characteristics of the wavelet bands of temporal and spatio-temporal decompositions of a video are then exploited for extracting representative frames for the video. A detailed discussion on the various possibilities for representing video in the 3D-DWT domain is presented along with the simulation results.

3.1 Video Representation Using Key Frames

Video representation with a key frame is a common procedure in the content-based video indexing and identification applications. In content-based video indexing, features of a video are extracted at the shot level [85]. Consider a video shot of size $N_1 \times N_2 \times N_3$, where $N_1 \times N_2$ is the dimension of each frame in the indexed set $\{f_k | k = 1, 2, \cdots, N_3\}$ representing the frames in the shot. In the video indexing applications, the shot is traditionally represented with the features extracted from a key frame f^{key}. f^{key} is supposed to contain the key information in the shot. As the frames in a shot represent the spatial characteristics of the shot, any one frame in the shot may be used as the key frame for the shot [86] [87]. The first frame (f_1), the middle frame ($f_{\frac{N_3}{2}}$) or the last frame (f_{N_3}) may be chosen as a key frame [76], [88].

The notion of key frame can be extended to video hashing [58]. The hash of f^{key} may be considered as a hash of the shot. The frames f_1 and f_{N_3} are not good choices here as they differ a lot from the intermediate frames. They may be significantly affected by temporal distortions. Therefore, $f_{\frac{N_3}{2}}$ is expected to perform better [58].

In video hashing, key frames are also selected by using a dissimilarity measure on the frames in the shot. Roover et al. [58] select f^{key} as the frame which is minimally different from the previous frame. Let $D_{k,k-1}$ represent the disparity between the frames f_k and f_{k-1}. To compute $D_{k,k-1}$, the authors use the ℓ_1 metric between the 64-bins luminance histograms of f_k and f_{k-1} according to:

$$D_{k,k-1} = \sum_{l=1}^{64} |\beta_k^l - \beta_{k-1}^l|, \tag{3.1}$$

where β_k^l represent the l^{th} value of the histogram of f_k. f^{key} is identified by the following two steps:

- Compute $D_{k,k-1}$ for each frame in the shot by using (3.1).
- Designate $f^{key} = f_z$, where $z = \arg\,(\min_{1<k\leq N_3} \{D_{k,k-1}\})$.

This key-frame selection technique is reported to be sensitive to noise.

In another method, Atrey et al. [79] designate a shot with multiple key frames based on a differential energy measurement. The authors divide each frame into M blocks of size 8×8. The differential energy $E_{x,y}$ between two frames f_x and f_y is computed as the weighted sum of the block-wise differential energies according to:

$$E_{x,y} = \sum_{l=1}^{M} \alpha_l \varepsilon_{x,y}^l, \tag{3.2}$$

where $\varepsilon_{x,y}^l$ is the difference energy of the l^{th} blocks in f_x and f_y. The weight α_l for the l^{th} block is decided based on the significance of the block. The key frames are selected by performing the following steps:

- Designate the first frame as the first key frame.
- Compute the differential energy of each subsequent frame with respect to the previous key frame by using (3.2). Include the frame as the next key frame for which this energy exceeds a suitably chosen threshold.
- Designate the last frame as the last key frame.

Note that the robustness of this key-frame selection technique against frame dropping is limited to the dropping of non-key frames.

One common drawback of the key-frame selection techniques is that they do not consider the temporal information in a video. Temporal information distinguishes a video from a series of random images. Hence, representative frames for video with embedded temporal information should intuitively perform better than the key-frames.

3.2 Temporally Informative Frames for Video Representation

Only a few works have investigated the problem of video representation by taking into account the temporal information in video. It has been recently explored that efficient representative frames for a video segment may be computed by considering the spatio-temporal information in the segment [8], [56], [57]. Three techniques for extracting representative frames for video segments are presented below.

(a) Representation based on Spatio-temporal Differencing of Block Means

Consider a video segment V_j of dimension $N_1 \times N_2 \times N_3$, where $N_1 \times N_2$ is dimension of each frame and N_3 is the number of frames in V_j. Let $\{f_k | k = 1, 2, \cdots, N_3\}$ represent the indexed set of the frames. In [8], Oostveen et al. divide the frames in V_j into blocks and compute the mean luminance of each block. A simple 2×2 spatio-temporal filter is applied on the mean luminance values of the blocks to eliminate the effects of the global level and the scales of the luminance values. The spatio-temporal filtering also reduces the temporal correlation of the block means. The steps for deriving the representative frames $f_k^{rep} | k = 1, 2, \cdots, N_3 - 1$ are listed in the following.

- Divide each frame in V_j into blocks of size $p \times p$ and compute the mean of the pixel-luminance values of each block. Let μ_k^l represent the mean of the l^{th} block of the k^{th} frame.
- Compute the spatio-temporal differences of the means in the consecutive frames according to: $f_k^{rep}(l) = (\mu_k^l - \mu_k^{l-1}) - (\mu_{k-1}^l - \mu_{k-1}^{l-1})$, where $f_k^{rep}(l)$ represent the luminance value at the location l of f_k^{rep}.

The difference of two adjacent frames represents the active motion components in the frames. The number of the representative frames is $N_3 - 1$ which is only one less than the number of frames in V_j. But, the representative frames are dimensionally $p \times p$ times smaller than than the original frames. For example, for a video in the CIF, the frame dimension is 288×352. With $p = 32$, the representative frames are of size 9×11. The representative frames are reported to have good robustness against noise. But, their sensitivity to frame dropping may not be acceptable.

(b) 3D-DCT based Representation

Coskun et al. [56] take into account both the temporal and spatial information in V_j during hash extraction. Before applying the 3D-DCT for hash extraction, the authors spatio-temporally subsam-

ple V_j to a normalised version $\overline{V}_j$ of dimension $\overline{N}_1 \times \overline{N}_2 \times \overline{N}_3$ to make the hash function resistant against the frame dropping and spatial resizing. Low-pass Gaussian filters are applied to both the temporal and spatial domains of V_j prior to the subsampling process to prevent aliasing. The frames in $\overline{V}_j$ may be considered as the representative frames for V_j. The following steps derive the frames.

- Apply low-pass Gaussian filters on the temporal and spatial domains of V_j.
- Subsample the filtered V_j to get $\overline{V}_j$.

Because of the low-pass filtering, the representative frames are expected to be robust against noise. The authors consider $\overline{V}_j$ of dimension $32 \times 32 \times 64$ in the reported work.

(c) Temporally Informative Representative Images for Representation

In this method of representation, Malekesmaeili et al. [57] represent V_j compactly by producing a composite representation of all the frames in V_j. The authors call the resulting representative frame a TIRI. Similar to [56], the algorithm in this case also smooth and subsample V_j before computing a representative frame f^{rep}. Let $\{\overline{f}_k | k = 1, 2, \cdots, \overline{N}_3\}$ represent the indexed set of the frames in $\overline{V}_j$, and $\overline{f}_k(n_1, n_2)$ represent the pixel luminance at the location (n_1, n_2) of the k^{th} frame. Weighted average of the frames $\overline{f}_k | k = 1, 2, \cdots, \overline{N}_3$ in $\overline{V}_j$ derives f^{rep} according to:

$$f^{rep}(n_1, n_2) = \sum_{k=1}^{\overline{N}_3} \alpha_k \overline{f}_k(n_1, n_2) \;\; ; \;\; n_1 = 1, 2, \cdots, \overline{N}_1, \; n_2 = 1, 2, \cdots, \overline{N}_2, \tag{3.3}$$

where the weight α_k associated with a frame may be chosen using different weighing functions. The steps for finding f^{rep} may be summerised as follows.

- Apply low-pass Gaussian filters on the temporal and spatial domains of V_j.
- Subsample the filtered V_j into $\overline{V}_j$.
- Compute f^{rep} from $\overline{V}_j$ using (3.3).

Due to the normalisation of V_j, f^{rep} is robust against spatio-temporal scaling. It is also reported to be robust against noise.

The above three representations involve spatio-temporal filtering and subsampling operations. These operations are inherent in the 3D-DWT. These inherent operations may be exploited to get the representative frames.

3.3 Content-Based Video Representation Using Temporal Bands of 3D-DWT

The luminance component of visual data is the most important component for the human visual system [14], [15]. For a compact representation, we consider extraction of representative frames from the luminance component of a video.

Consider a video V divided into N GOFs $G_1, G_2, \cdots, G_N$, each of size $N_1 \times N_2 \times N_3$. The luminance component of each GOF may be decomposed along the temporal direction by applying one-dimensional DWT up to a level $u \leq \hat{u}$, where $\hat{u} = \lceil \log_2 N_3 \rceil$ and $\lceil \cdot \rceil$ rounds a number to the nearest larger integer. Let tuL represent the low-pass band and $tuH, t(u-1)H, \cdots, t1H$ represent the high-pass bands of temporal decomposition of a GOF G_j. These bands carry information about the temporal content of G_j at different frequency regions. The information contained in some of the bands are exploited in the following for extracting representative frames for G_j.

3.3.1 Representative Frames from Temporal Low-pass Bands

The low-pass band of temporal decomposition of G_j provides a compact representation of G_j. Therefore, it is the most important band among the all temporal bands. For concise representation, we individually examine the temporal low-pass bands $t\hat{u}L$ and $t(\hat{u}-1)L$ of G_j for representing G_j.

Case-1: Representation by the frame in $t\hat{u}L$

The low-pass band $t\hat{u}L$ at the full-level of temporal decomposition contains only one frame $f^{t\hat{u}L}$. This low-pass frame carries information from every frame in G_j and is a summary of the spatio-temporal content of G_j. As it is derived by the temporal low-pass filtering, the high frequency noise present in G_j is smoothed out during the process of deriving the frame. Therefore, the content of $f^{t\hat{u}L}$ is expected to be robust against noise. It should also show robustness against frame dropping because of the low-pass filtering operation. Again, the lossy video-compression schemes (scalable or otherwise) retain the low-pass information in video data during compression. As the content of $f^{t\hat{u}L}$ is at the lowest frequency region, it should be robust against compressing the GOF. These considerations suggest that the $f^{t\hat{u}L}$ may be considered as a representative frame for G_j.

Case-2: Representation by the frames in $t(\hat{u}-1)L$

The low-pass band $t(\hat{u}-1)L$ at the $(\hat{u}-1)^{\text{th}}$ level of temporal decomposition contains more high-frequency information in comparison to $t\hat{u}L$ band. It contains two frames $f_1^{t(\hat{u}-1)L}$ and $f_2^{t(\hat{u}-1)L}$. Similar to the case of $f^{t\hat{u}L}$, the contents of $f_1^{t(\hat{u}-1)L}$ and $f_2^{t(\hat{u}-1)L}$ are also expected to have robustness against noise, frame dropping and lossy compression. Representation of G_j may be also possible with these two frames.

3.3.2 Representative Frames from Temporal High-pass Bands

The high-pass bands of temporal decomposition of a GOF inherit the motion or temporal characteristics of the GOF. In other words, they represent the degree of homogeneity of the frames in G_j [79]. For example, the energy of the frames in a high-pass band of a fast GOF is more than that of the corresponding frames of a slow GOF. This is because the frames in the fast GOF are less homogeneous than the frames in the slow GOF. Hence, the frames in selected high-pass bands should be useful for representing G_j. The high-pass bands at the lower levels of decomposition contain the motion information better than those at the higher levels. But, the bands at lower levels of decomposition are more vulnerable to frame dropping and noise. Also, the bands at higher levels of decomposition are ideal for compact representation due to their smaller sizes. Therefore, the two high-pass bands $t\hat{u}H$ and $t(\hat{u}-1)H$ are individually considered here for representing G_j.

Case-3: Representation by the frame in $t\hat{u}H$

This high-pass band contains one frame $f^{t\hat{u}H}$. This frame is a compact representation of the motion information in G_j. Since it is at a very low-frequency region, its content should be sufficiently robust to noise, frame dropping and lossy compression. Therefore, $f^{t\hat{u}H}$ may be another option for representing G_j.

Case-4: Representation by the frames in $t(\hat{u}-1)H$

The high pass band $t(\hat{u}-1)H$ contains information at a higher frequency region in comparison to the band $t\hat{u}H$. It contains two frames $f_1^{t(\hat{u}-1)H}$ and $f_2^{t(\hat{u}-1)H}$. The contents of these two frames are also expected to have sufficient robustness against noise, frame dropping and lossy compression. Hence, representation of G_j may be possible with these frames.

3.3.3 Similarity Measure and Representation Performance

Different distance measures for images have been applied in literatures to measure the similarity between key / representative frames for different applications. In video hashing, the ℓ_1 metric has been used on the 64-bins luminance histograms of key frames for concluding similarity between frames [58]. Panchanathan et al. in [87] introduce a normalised ℓ_1 metric at the image block level to measure the similarity between images in the DWT domain.

(a) Perceptual Blocks and Binarisation

In the proposed work, the similarity between two GOFs is concluded on the basis of the similarity between the corresponding representative frames for the GOFs. The two GOFs are declared similar when each pair of the corresponding representative frames are similar. To take into account the local variations in the contents of the GOFs, the similarity between two representative frames are computed at the block level. We call these blocks *perceptual block*s. A perceptual block of size $p \times p$ represents a pixel volume of size $p \times p \times N_3$ in a raw GOF. The block sizes are chosen in such a way that the blocks are able to detect perceptually important differences in the contents of distinct GOFs. The representative frames for the two GOFs are divided into perceptual blocks and blocks at the same location in the corresponding representative frames are to be compared.

The wavelet coefficients in each perceptual block are binarised by means of a thresholding operation. The purpose of binarisation is two fold: (i) the resulting binary frames contain information about the contents. At the same time, the binary frames are less sensitive to the perceptually unimportant changes [56], and (ii) The compression of the contents becomes easier when they have binary representation.

The median or mean [89] of the wavelet coefficients in a perceptual block may be considered as the threshold for binarisation of the coefficients. Suppose that the perceptual blocks in the representative frames are identically and independently distributed. Binarisation by median thresholding results in equal numbers of 1's and 0's in the binarised data. In this case, we can assign a probability of 0.5 for each bit to take the value of 1 and the statistical modelling of binary data become easy. One disadvantage of the median thresholding is the complexity in finding the median. The median is a non-linear operation requiring sorting of the data. A simple alternative to the median thresholding is the mean thresholding. We propose to use thresholding about local mean to binarise the contents of the perceptual blocks. In the case of mean thresholding, the values of 1 and 0 are not equally likely

except for a symmetrical distribution of data. Though the high-pass wavelet bands in the case of image are known to follow a symmetrical distribution like the Laplacian or the Generalised Gaussian distribution [90], such models are not satisfied by the low-pass wavelet band. Our experimental observations on a number of binarised frames show this probability in the range [0.42 0.57]. We assume that each bit resulting from mean thresholding takes values of 1 and 0 with the equal probability of 0.5.

Consider two GOFs G_x and G_y with the index sets of representative frames $\{f_{x,k}^{rep}|k = 1, 2, \cdots, Q\}$ and $\{f_{y,k}^{rep}|k = 1, 2, \cdots, Q\}$ respectively. Each representative frame is partitioned into an index set of M perceptual blocks of size $p \times p$, given by $\{B_k^l|l = 1, 2, \cdots, M\}$.

Consider a representative frame $f_{x,k}^{rep}$. Let μ_k^l be the mean of the wavelet coefficients in the l^{th} perceptual block B_k^l given by

$$\mu_k^l = \frac{1}{p \times p} \sum_{(n_1,n_2)} \sum_{\in B_k^l} f_{x,k}^{rep}(n_1, n_2). \tag{3.4}$$

Let $\hat{f}_{x,k}^{rep}$ be the binary version of $f_{x,k}^{rep}$. Each coefficient in B_k^l is thresholded to derive the binary frame $\hat{f}_{x,k}^{rep}$ according to:

$$\hat{f}_{x,k}^{rep}(n_1, n_2) = \begin{cases} 1 & ; \quad \text{if } f_{x,k}^{rep}(n_1, n_2) \geq \mu_k^l \\ 0 & ; \quad \text{otherwise,} \end{cases} \tag{3.5}$$

where $(n_1, n_2) \in B_k^l$ and $l = 1, 2, \cdots, M$.

(b) Similarity Measure

The content difference in a pair of corresponding pixel volumes can potentially change the meaning of G_x and G_y. For example, consider the GOF of the first 32 frames from the Mobile video. The semantic information in the GOF changes completely when the calender-month mark '1' in the frames is changed to '2' by replacing one $32 \times 32 \times 32$ pixel-volume in the GOF. The first frames from the original and modified GOFs are shown in Figure 3.1. The similarity measure should sense such content difference between two GOFs at the block level. The two representative frames $f_{x,k}^{rep}$ and $f_{y,k}^{rep}$ are declared similar only when all the pairs of the corresponding perceptual blocks are similar.

(a) (b)

Fig. 3.1: The first frames from the (a) original Mobile GOF (b) modified Mobile GOF.

It is mentioned in Chapter 1 that the Hamming distance is a popular distance measure used to compare two binary strings. In [8], Oostveen et al. use it to compare two binary frames. We also propose to use the Hamming distance to compare the corresponding binarised perceptual blocks in $\hat{f}^{rep}_{x,k}$ and $\hat{f}^{rep}_{y,k}$. The Hamming distance between the l^{th} binarised perceptual blocks in the two frames is given by

$$d^l_k = \sum_{(n_1,n_2)} \sum_{\in B^l_k} \hat{f}^{rep}_{x,k}(n_1,n_2) \oplus \hat{f}^{rep}_{y,k}(n_1,n_2), \tag{3.6}$$

where the symbol $\oplus$ represents the Exclusive-OR operation [91] on two bits. For two GOFs to be perceptually similar, the Hamming distance between each pair of the corresponding binarised perceptual blocks should remain within a threshold. In other words, the maximum of the Hamming distances over these blocks should remain within a threshold. The similarity between G_x and G_y can be measured in terms of the *similarity value* $S(G_x, G_y)$ in the range [0 1] defined as

$$S\left(G_x, G_y\right) = 1 - \max_{1\le k\le Q}\left(\max_{1\le l\le M}\left(\frac{d^l_k}{p\times p}\right)\right), \tag{3.7}$$

where 'max' is the maximisation operator and $\frac{d^k_l}{p\times p}$ is the normalised Hamming distance. In other words, G_x and G_y are similar if

$$S\left(G_x, G_y\right) \ge T_2, \tag{3.8}$$

where the threshold T_2 is chosen suitably. The condition in (3.8) is equivalent to

$$\max_{1\leq k\leq Q}\left(\max_{1\leq l\leq M}\left(d_k^l\right)\right)\leq T_1. \tag{3.9}$$

The threshold T_1 is related to T_2 according to:

$$T_1 = p\times p(1-T_2). \tag{3.10}$$

(c) Representation Performance

Let C_1 and C_2 be respectively the number of content-wise similar GOF pairs declared as similar and the number of similar GOF pairs under test. The *representation performance for content similarity* (R_1) is computed according to:

$$R_1 = \frac{C_1}{C_2}. \tag{3.11}$$

Similarly, the *representation performance for content dissimilarity* (R_2) is obtained by using

$$R_2 = \frac{C_3}{C_4}, \tag{3.12}$$

where C_3 and C_4 are respectively the number of content-wise dissimilar GOF pairs declared as dissimilar and the number of dissimilar GOF pairs under test. The overall representation performance is measured in terms of *performance index* (PI) given by

$$PI = \frac{C_1+C_3}{C_2+C_4}. \tag{3.13}$$

3.3.4 Selection of the Threshold T_2

For verifying the similarity of G_x and G_y using (3.9) or (3.8), a suitable value for T_1 or T_2 is to be determined. The M pairs of the corresponding binarised perceptual blocks in $\hat{f}_{x,k}^{rep}$ and $\hat{f}_{y,k}^{rep}$ derive M Hamming distances. The similarity between G_x and G_y is decided on the basis of the maximum Hamming distance out of the total $Q\times M$ Hamming distances obtained from the Q pairs of the corresponding representative frames in $\{f_{x,k}^{rep}|k=1,2,\cdots,Q\}$ and $\{f_{y,k}^{rep}|k=1,2,\cdots,Q\}$.

Assume the GOFs G_x and G_y to be distinct. Let the random variables $d_1^1, d_1^2, \cdots, d_1^M, d_2^1, \cdots, d_{Q-1}^M$, $d_Q^1, \cdots, d_Q^M$ represent the Hamming distances of the corresponding binarised perceptual blocks and $\gamma = \max\left(d_1^1, d_1^2, \cdots, d_1^M, d_2^1, \cdots, d_{Q-1}^M, d_Q^1, \cdots, d_Q^M\right)$. Each of $d_1^1, d_1^2, \cdots, d_1^M, d_2^1, \cdots, d_{Q-1}^M, d_Q^1, \cdots, d_Q^M$ has a cumulative distribution function (CDF) [89] $F_d(q)$, $q = 0, 1, \cdots, p \times p$. Assume that all the bits in a binarised perceptual block are independent. Under this assumption, each d_k^l is a *binomial random variable* [89] with $F_d(q)$ given by [56]

$$F_d(q) = \sum_{i=0}^{q} \binom{p^2}{i} (0.5)^{p \times p}, \tag{3.14}$$

where $\binom{p^2}{i} = \frac{p^2!}{i!(p^2-i)!}$ represents the combination operation. Let $F_\gamma(q)$ be the CDF of γ at a point $q = 0, 1, \cdots, p \times p$. Then, $F_\gamma(q)$ is given by

$$\begin{aligned} F_\gamma(q) &= P(\gamma \leq q) \\ &= P\left(d_1^1 \leq q, d_1^2 \leq q, \cdots, d_1^M \leq q, \cdots, d_Q^1 \leq q, d_Q^2 \leq q, \cdots, d_Q^M \leq q\right) \\ &= \left(F_d(q)\right)^{Q \times M}. \end{aligned} \tag{3.15}$$

The probability of a maximum Hamming distance $\gamma = q$ is obtained as

$$\begin{aligned} P(\gamma = q) &= F_\gamma(q) - F_\gamma(q-1) \\ &= \begin{cases} \left(F_d(q)\right)^{Q \times M} - \left(F_d(q-1)\right)^{Q \times M} & ; \quad \text{for } 1 \leq q \leq p \times p \\ \left(F_d(q)\right)^{Q \times M} & ; \quad \text{for } q = 0. \end{cases} \end{aligned} \tag{3.16}$$

The mean μ_γ of the maximum Hamming distances is obtained according to:

$$\begin{aligned} \mu_\gamma &= \sum_{q=1}^{p \times p} P(\gamma = q) q \\ &= \sum_{q=1}^{p \times p} \left(\left(F_d(q)\right)^{Q \times M} - \left(F_d(q-1)\right)^{Q \times M}\right) q. \end{aligned} \tag{3.17}$$

Assume that two similar GOFs have ideally zero Hamming distance. The threshold T_1 in (3.9) may be set at

$$T_1 = \frac{0 + \mu_\gamma}{2} = \frac{\mu_\gamma}{2} \tag{3.18}$$

and T_2 can be computed by using the expression in (3.10).

Among the four cases of representation considered in Subsections 3.3.1 and 3.3.2, each of the $t\hat{u}L$ and $t\hat{u}H$ bands have one representative frame. It is observed later in Subsection 3.3.6 that for GOFs in the CIF format, 32×32 perceptual blocks in the representative frames can fairly resolve significant differences in the contents of distinct GOFs. The representative frame in the $t\hat{u}L$ or $t\hat{u}H$ band has 99 perceptual blocks. With $p \times p = 32 \times 32$ and $M = 99$, the probabilities of the maximum Hamming distances at $q = 0, 1, \cdots, 1024$ are computed according to the model in (3.16). The plot of these probabilities at intervals of 8 units is shown in Figure 3.2(a). Using (3.17), the mean of the maximum Hamming distances is computed to be $\mu_\gamma = 552.045$.

Therefore, using (3.18), T_1 may be chosen as

$$T_1 = \frac{\mu_\gamma}{2} = \frac{552.045}{2} = 276.023. \tag{3.19}$$

Therefore, T_2 for the $t\hat{u}L$ or $t\hat{u}H$ bands is computed by using the expression in (3.10) as

$$T_2 = 1 - \frac{T_1}{p \times p} = 1 - \frac{276.023}{32 \times 32} = 0.730. \tag{3.20}$$

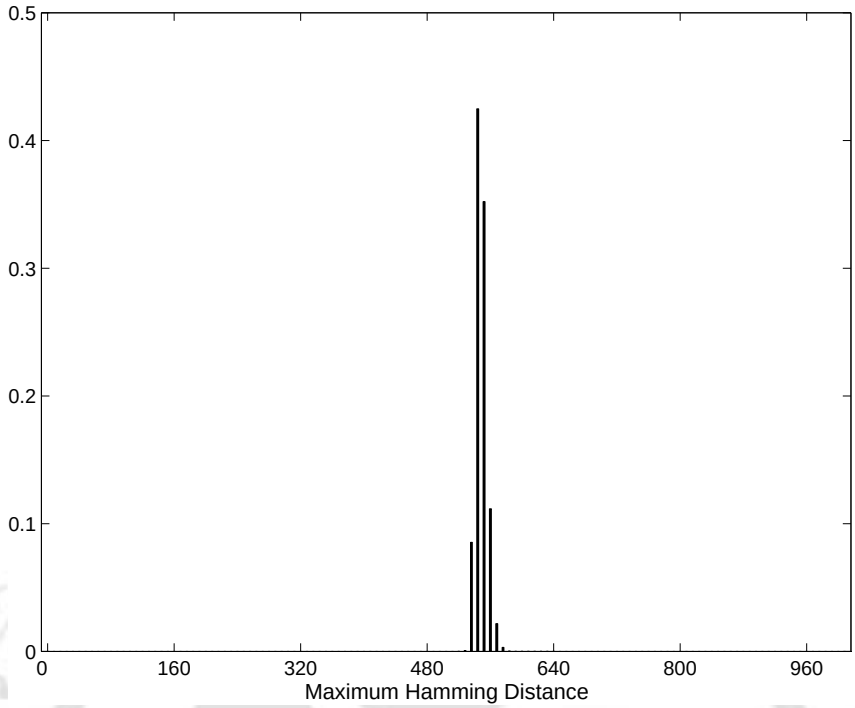
The $t(\hat{u}-1)L$ and $t(\hat{u}-1)H$ bands have two representative frames each. For a GOF in the CIF format, the two representative frames in each of the $t(\hat{u}-1)L$ and $t(\hat{u}-1)H$ bands have 198 perceptual blocks. With $p \times p = 32 \times 32$ and $M = 198$, the probabilities of the maximum Hamming distances at $q = 0, 1, \cdots, 1024$ are computed by using (3.16). The plot of these probabilities at intervals of 8 units is shown in Figure 3.2(b). In this case, the mean of the maximum Hamming distances by using (3.17) is computed to be $\mu_\gamma = 555.858$. Accordingly, by using (3.18), T_1 is obtained as

$$T_1 = \frac{\mu_\gamma}{2} = \frac{555.858}{2} = 277.929. \tag{3.21}$$

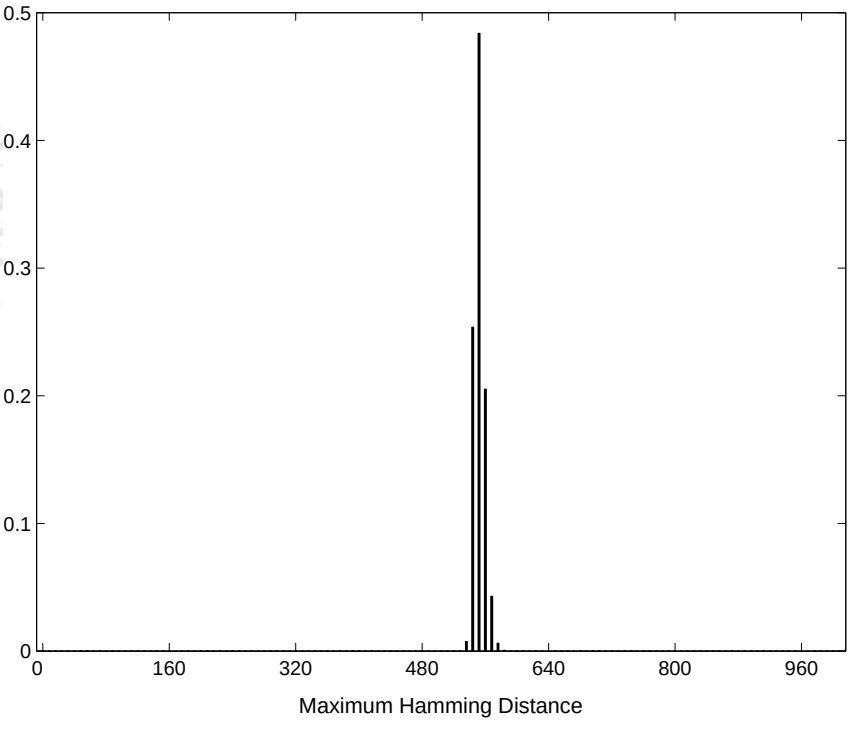
Thus, the threshold T_2 for the $t(\hat{u}-1)L$ and $t(\hat{u}-1)H$ bands is:

$$T_2 = 1 - \frac{T_1}{p \times p} = 1 - \frac{277.929}{32 \times 32} = 0.729. \tag{3.22}$$

Table 3.1 shows the values for T_1 and T_2 according to the statistical model when each of the four temporal bands are used for representation. It can be seen that the values $T_2 = 0.730$ for the $t\hat{u}L$



(a) $p \times p = 32 \times 32$, $M = 99$



(b) $p \times p = 32 \times 32$, $M = 198$

Fig. 3.2: The probabilities of the maximum Hamming distances at intervals of 8 units for distinct GOFs in the CIF format represented with the frame(s) in the (a) $t\hat{u}L$ or $t\hat{u}H$ band (b) $t(\hat{u}-1)L$ or $t(\hat{u}-1)H$ band

and $t\hat{u}H$ bands and $T_2 = 0.729$ for the $t(\hat{u}-1)L$ and $t(\hat{u}-1)H$ bands are very close. Hence, in all the four cases, we consider $T_2 = 0.73$.

Table. 3.1: The values obtained from the statistical model for T_1 and T_2 when GOFs in the CIF format are represented with temporal wavelet bands ($p \times p = 32 \times 32$)

Temporal wavelet band		T_1	T_2
Low-pass	$t\hat{u}L$	276.023	0.730
	$t(\hat{u}-1)L$	277.929	0.729
High-pass	$t\hat{u}H$	276.023	0.730
	$t(\hat{u}-1)H$	277.929	0.729

3.3.5 Temporal Wavelet Bands for Representation: an Analysis

In the following, we discuss the strong points of using the frames in the $t\hat{u}L$, $t(\hat{u}-1)L$, $t\hat{u}H$ or $t(\hat{u}-1)H$ band of a GOF for representing the GOF.

Sensitivity and size of perceptual block: The sensitivity of a representative frame can be defined as the smallest perceptual difference that the frame can resolve. Here, the larger the perceptual-block size $p \times p$ is, the lower is the sensitivity of the representative frame to tiny content differences. But, when the perceptual blocks are very small, the representative frame ceases to be robust against the content-preserving operations. As discussed later in Subsection 3.3.6, for a GOF in the CIF format with 32 frames, 32×32 perceptual blocks in the representative frames can fairly resolve significant differences.

In addition, the perceptual blocks may fail to recognize small differences in the contents of two representative frames when the differences are present at the block boundaries distributing over multiple blocks. To overcome this limitation, an overlapping perceptual blocks may be considered.

Robustness against temporal scaling by wavelet transform: The contents of the four bands $t\hat{u}L$, $t(\hat{u}-1)L$, $t\hat{u}H$ and $t(\hat{u}-1)H$ of the GOF are naturally robust against the scalability of the wavelet transform along the temporal direction. While the contents of the bands $t\hat{u}L$ and $t\hat{u}H$ are robust up to the full-level of wavelet-based temporal scalability, the contents of the other two bands are robust up to one level less than the full level.

Robustness against MPEG compression: The lossy video coders, including the MPEG coders, retain the low-frequency components of the visual data. The bands $t\hat{u}L$, $t(\hat{u}-1)L$, $t\hat{u}H$ and $t(\hat{u}-1)H$ are in the temporal low-frequency region. Therefore, the contents of the frames in these bands are likely to be robust against the MPEG compression.

Robustness against spatial averaging: The spatial averaging is used in noise smoothing and low-pass filtering operations [92]. During verifying the similarity between two representative frames, the frames are divided into perceptual blocks. The wavelet coefficients in each perceptual blocks are thresholded with respect to the mean of the coefficients. Hence, the representative frames in all the four cases are expected to perform against the spatial averaging operations on raw video frames.

Robustness against brightness and contrast modifications: During a brightness modification, the change is constant within each raw frame. As a result, the frames in the temporal wavelet bands also experience constant change. Therefore, brightness modifications should not affect the similarity between the corresponding representative frames for two GOFs. Again, because of high spatio-temporal correlation of video data, the contrast modification does not affect much the dynamic relationships of the coefficients in a block of a representative frame. Therefore, the contents of the representative frames in all the four cases are expected to be robust against contrast modifications. But, these frames may fail to serve when the brightness and contrast changes are beyond saturation.

Robustness against frame-rate reductions: In a raw video, the frame rate is reduced by dropping frames either at a regular interval or at random. As the $t\hat{u}L$, $t(\hat{u}-1)L$, $t\hat{u}H$ and $t(\hat{u}-1)H$ bands are in the temporal low-frequency region, their contents are likely to be robust against the frame-rate reductions in the pixel domain. In the wavelet domain, the frame rate is reduced by truncating temporal high-pass bands selectively. The $t\hat{u}L$ and $t(\hat{u}-1)L$ bands remain available when the high-pass bands up to the levels $\hat{u}$ and $\hat{u}-1$ are truncated respectively. In the cases of the $t\hat{u}H$ and $t(\hat{u}-1)H$ bands, the high-pass bands up to the levels $\hat{u}-1$ and $\hat{u}-2$ can be truncated.

3.3.6 Experimental Observations and Analysis I

To study the content-representation performance of the proposed temporal bands, 14 test videos in the CIF format with a frame rate of 30 frames/second (fps) are considered for experimentation. These videos are Akiyo, Antibes, Bike, Cheer, Coastguard, Container, Football, Foreman, Garden, Mobile, Mosaic, News, Stefan and Tempete. Four GOFs, each of 32 frames, are used from each of

the videos. That is, altogether 56 GOFs are used from the 14 test videos. A GOF of 32 frames is a video segment of approximately one second duration and is chosen to obtain representative frames per second basis. The representation for one second duration may be desirable, for example, when video segments of duration as low as one second are to be identified [8]. The GOFs are decomposed temporally up to the fifth and the fourth levels separately. The Haar filters are used in the temporal decomposition as they can handle smaller number of frames with less boundary problems. They also offer a good trade-off between the delay and the energy compaction [51]. The temporal bands $t5L$, $t4L$, $t5H$ and $t4H$ are individually used for representation according to the four cases discussed in Subsections 3.3.1 and 3.3.2.

(a) Selection of the Size of the Perceptual Blocks

The perceptual block size $p \times p$ is decided in a manner that the local distinctiveness in the contents of GOFs can be detected from the contents of their temporal bands. As discussed earlier, the perceptual blocks should be large enough to contain significant perceptual information and small enough to be affected by any perceptual difference in the contents of the GOFs. Normally, a perceptual block size of 16×16 may be chosen in the case of raw image [12]. But, in our case, it is experimentally observed that 32×32 perceptual blocks in the temporal wavelet bands carry sufficient information for discrimination. Hence, the representative frames are divided into 32×32 perceptual blocks. The experimental results are presented below.

(b) Dissimilarity of GOF-wise Content

We first study the performances of the four cases of representations by the temporal bands $t5L$, $t4L$, $t5H$ and $t4H$ on different videos separately. Since representation at the GOF level is considered, the representative frames for consecutive GOFs in a video should be dissimilar. The GOFs from the Coastguard, Container, Foreman and Mobile videos are considered for demonstration. In each case of representation, the similarity value for G_1 and G_2 from each video is computed using (3.7). The similarity values for G_2 and G_3 and for G_3 and G_4 from each video are also computed. The results are shown in Table 3.2. As expected, all the similarity values are below $T_2 = 0.73$.

Table. 3.2: The similarity values for the adjacent GOFs from the Coastguard, Container, Foreman and Mobile videos by using the $t5L$, $t4L$, $t5H$ and $t4H$ bands for representation

(a) $S(G_1, G_2)$

GOF	Temporal low-pass Band		Temporal high-pass Band	
	$t5L$	$t4L$	$t5H$	$t4H$
Coastguard	0.2236	0.2783	0.2168	0.2207
Container	0.4824	0.4023	0.2920	0.3105
Foreman	0.1006	0.0508	0.0762	0.1504
Mobile	0.2725	0.2363	0.1914	0.3447

(b) $S(G_2, G_3)$

GOF	Temporal low-pass Band		Temporal high-pass Band	
	$t5L$	$t4L$	$t5H$	$t4H$
Coastguard	0.2461	0.2363	0.2510	0.2061
Container	0.4766	0.3394	0.3633	0.3525
Foreman	0.1719	0.0781	0.0664	0.1455
Mobile	0.3008	0.3340	0.2285	0.3613

(c) $S(G_3, G_4)$

GOF	Temporal low-pass Band		Temporal high-pass Band	
	$t5L$	$t4L$	$t5H$	$t4H$
Coastguard	0.3330	0.1797	0.1738	0.2051
Container	0.4717	0.3564	0.2090	0.2246
Foreman	0.2168	0.0732	0.2002	0.1436
Mobile	0.3467	0.3252	0.2100	0.3057

(c) Similarity of Contents under Content-Preserving Operations

It is also required that the representative frames for two GOFs from distinct videos should be very dissimilar. The dissimilarity should be preserved even after content-preserving operations on the GOFs. At the same time, the contents of the representative frames for a GOF should not change much due to the content-preserving operations on the GOF. For demonstration, the first GOFs from the Coastguard, Container, Foreman and Mobile videos are subjected to the following content-preserving operations.

i. MPEG compression: To examine the effect of the lossy compression on the contents of the representative frames, the GOFs are compressed using an MPEG-2 coder at the bit-rate of 64kbps and decompressed. The decompressed GOFs are considered for experimentation.

ii. Spatial averaging: The spatial averaging masks of sizes 3×3 and 5×5 are applied to the frames of the GOFs. This derives two processed GOFs per GOF.

iii. Brightness modification: The brightness of each frame in the GOFs is increased / decreased by

50% of the original frame intensity.

iv. Contrast modification: For enhancing the contrast of the GOFs, the histogram equalisation method for contrast enhancement is used.

v. Frame dropping: The frame rate is reduced from 30 fps to 15 fps by dropping frames in the GOFs in regular and random manner.

vi. AWGN addition: The GOFs are corrupted with the zero-mean AWGN with variance of 5, 10 and 20. Here, three GOFs per GOF are derived.

Hence, for each of the four *original* GOFs, we obtain a group of 12 similar GOFs including the original GOF and 11 processed GOFs. All the GOFs, except the ones derived after frame dropping, are temporally decomposed up to the fifth and fourth levels separately. The GOFs derived after the frame-dropping operations are decomposed up to the levels four and three separately. For each case of representation, the similarity value for the processed GOFs and each original GOF is computed by using (3.7). An original GOF is equivalent to the result of performing the identity operation on the original GOF.

Table 3.3 presents the similarity values when the wavelet frame in the $t5L$ band is used for representation. As can be observed in the table, similarity values smaller than $T_2 = 0.73$ are obtained for similar GOFs in a few instances. These values are underlined. From the table, the robustness of the content of the $t5L$ band is found to be poor against the spatial averaging using the 5×5 mask and AWGN of high variances. The content of the band is also sensitive to the MPEG-2 compression and brightness and contrast modifications. For dissimilar GOFs, all the similarity values are below 0.73.

When the frames in the temporal low-pass band $t4L$ are used for representation, the observed similarity values are presented in Table 3.4. The similarity values for similar GOFs that are below $T_2 = 0.73$ are underlined. As observed here, the content of this band is sensitive to the MPEG-2 compression, spatial averaging using the 5×5 mask, brightness and contrast modifications and AWGN of high variances. But, the dissimilar GOFs have similarity values well below 0.73.

The similarity values when the high-pass bands $t5H$ and $t4H$ are used for representation are respectively shown in Table 3.5 and Table 3.6. The similarity values for similar GOFs, which are smaller than $T_2 = 0.73$ are underlined. These results suggest that the contents of the $t5H$ and $t4H$ bands are more sensitive to the content-preserving operations than those of the $t5L$ and $t4L$ bands. It appears that the $t5H$ and $t4L$ bands may not perform adequately when used to represent GOFs.

Table. 3.3: The similarity values for the original GOFs and the GOFs derived after the content-preserving operations when the $t5L$ band is used for representation

GOF	Content-preserving Operation	GOF			
		Coastguard	Container	Foreman	Mobile
Coastguard	Identity	1	0.2656	0.1426	0.2129
	MPEG-2 Compression (64kbps)	0.6689	0.2637	0.1719	0.2158
	Spatial averaging (3×3)	0.8590	0.2631	0.1410	0.2178
	Spatial averaging (5×5)	0.6211	0.1904	0.1747	0.1572
	Brightness modification (+50%)	0.8916	0.2656	0.1426	0.2051
	Brightness modification (-50%)	0.7979	0.2500	0.2451	0.1917
	Contrast modification (HE)	0.7559	0.2637	0.1611	0.1865
	Frame dropping (50%, regular)	0.8818	0.2744	0.1494	0.2188
	Frame dropping (50%, random)	0.8952	0.2843	0.1607	0.2095
	AWGN addition ($\sigma^2 = 5$)	0.7949	0.3286	0.2277	0.2361
	AWGN addition ($\sigma^2 = 10$)	0.7596	0.3398	0.2705	0.2490
	AWGN addition ($\sigma^2 = 20$)	0.5378	0.3831	0.2542	0.2755
Container	Identity	0.2656	1	0.1230	0.1797
	MPEG-2 Compression (64kbps)	0.2441	0.6104	0.1055	0.1777
	Spatial averaging (3×3)	0.2647	0.8733	0.1272	0.1803
	Spatial averaging (5×5)	0.2237	0.6893	0.1863	0.2846
	Brightness modification (+50%)	0.3047	0.7969	0.1230	0.1729
	Brightness modification (-50%)	0.2686	0.7559	0.0625	0.1797
	Contrast modification (HE)	0.3125	0.5430	0.0713	0.1709
	Frame dropping (50%, regular)	0.2656	0.9541	0.1230	0.1797
	Frame dropping (50%, random)	0.2385	0.9326	0.1044	0.1565
	AWGN addition ($\sigma^2 = 5$)	0.3672	0.8307	0.1250	0.1797
	AWGN addition ($\sigma^2 = 10$)	0.4213	0.7875	0.1250	1797
	AWGN addition ($\sigma^2 = 20$)	0.4355	0.6137	0.2346	0.2251
Foreman	Identity	0.1426	0.1230	1	0.1982
	MPEG-2 Compression (64kbps)	0.1396	0.1289	0.8359	0.1807
	Spatial averaging (3×3)	0.1309	0.1260	0.8936	0.1992
	Spatial averaging (5×5)	0.1730	0.1991	0.6782	0.1705
	Brightness modification (+50%)	0.2598	0.0625	0.5615	0.1973
	Brightness modification (-50%)	0.1338	0.1260	0.8750	0.1982
	Contrast modification (HE)	0.1377	0.1240	0.8096	0.2012
	Frame dropping (50%, regular)	0.1631	0.1250	0.9180	0.1777
	Frame dropping (50%, random)	0.1763	0.1046	0.8753	0.1519
	AWGN addition ($\sigma^2 = 5$)	0.1739	0.1472	0.8250	0.2002
	AWGN addition ($\sigma^2 = 10$)	0.1748	0.1318	0.7574	0.2041
	AWGN addition ($\sigma^2 = 20$)	0.1864	0.1487	0.5527	0.2278
Mobile	Identity	0.2129	0.1797	0.1982	1
	MPEG-2 Compression (64kbps)	0.2012	0.1670	0.2012	0.8252
	Spatial averaging (3×3)	0.2103	0.1799	0.1973	0.8944
	Spatial averaging (5×5)	0.2853	0.2206	0.1958	0.7444
	Brightness modification (+50%)	0.2041	0.1719	0.1992	0.8994
	Brightness modification (-50%)	0.2129	0.1846	0.1963	0.7197
	Contrast modification (HE)	0.2070	0.1807	0.1934	0.8994
	Frame dropping (50%, regular)	0.2090	0.1768	0.1982	0.9541
	Frame dropping (50%, random)	0.2058	0.1866	0.1887	0.9438
	AWGN addition ($\sigma^2 = 5$)	0.2076	0.1745	0.2047	0.8460
	AWGN addition ($\sigma^2 = 10$)	0.2070	0.1816	0.2031	0.7939
	AWGN addition ($\sigma^2 = 20$)	0.1968	0.1549	0.1933	0.7012

Table. 3.4: The similarity values for the original GOFs and the GOFs derived after the content-preserving operations when the $t4L$ band is used for representation

GOF	Content-preserving Operation	GOF			
		Coastguard	Container	Foreman	Mobile
Coastguard	Identity	1	0.3105	0.1670	0.2607
	MPEG-2 Compression (64kbps)	0.6387	0.2988	0.1929	0.2236
	Spatial averaging (3×3)	0.8720	0.3133	0.1703	0.2599
	Spatial averaging (5×5)	0.6338	0.3242	0.1172	0.2246
	Brightness modification (+50%)	0.9053	0.3105	0.1670	0.2607
	Brightness modification (-50%)	0.8555	0.3027	0.1543	0.2451
	Contrast modification (HE)	0.8311	0.2959	0.1699	0.2588
	Frame dropping (50%, regular)	0.8594	0.3213	0.1729	0.2480
	Frame dropping (50%, random)	0.8777	0.3307	0.1843	0.2053
	AWGN addition ($\sigma^2 = 5$)	0.7518	0.3332	0.1754	0.2857
	AWGN addition ($\sigma^2 = 10$)	0.5518	0.3613	0.2109	0.3193
	AWGN addition ($\sigma^2 = 20$)	0.5361	0.4530	0.2648	0.3395
Container	Identity	0.3105	1	0.1367	0.2529
	MPEG-2 Compression (64kbps)	0.2920	0.5635	0.1387	0.2471
	Spatial averaging (3×3)	0.3100	0.8237	0.1382	0.2533
	Spatial averaging (5×5)	0.3029	0.6266	0.1358	0.2590
	Brightness modification (+50%)	0.2490	0.8164	0.1543	0.2461
	Brightness modification (-50%)	0.3174	0.8135	0.1250	0.2529
	Contrast modification (HE)	0.2480	0.6104	0.1738	0.2432
	Frame dropping (50%, regular)	0.3125	0.9395	0.1367	0.2520
	Frame dropping (50%, random)	0.3100	0.9271	0.1528	0.2668
	AWGN addition ($\sigma^2 = 5$)	0.3941	0.6429	0.1685	0.2847
	AWGN addition ($\sigma^2 = 10$)	0.4518	0.5625	0.1494	0.2510
	AWGN addition ($\sigma^2 = 20$)	0.4823	0.5471	0.2221	0.2693
Foreman	Identity	0.1670	0.1367	1	0.3145
	MPEG-2 Compression (64kbps)	0.1709	0.1367	0.7090	0.3076
	Spatial averaging (3×3)	0.1680	0.1338	0.8633	0.3076
	Spatial averaging (5×5)	0.1694	0.1381	0.6415	0.3189
	Brightness modification (+50%)	0.3477	0.0625	0.4395	0.3125
	Brightness modification (-50%)	0.1670	0.1406	0.8174	0.3145
	Contrast modification (HE)	0.1680	0.1357	0.7900	0.3184
	Frame dropping (50%, regular)	0.1689	0.1328	0.9219	0.3086
	Frame dropping (50%, random)	0.1689	0.1457	0.9004	0.2849
	AWGN addition ($\sigma^2 = 5$)	0.1405	0.3237	0.6903	0.3656
	AWGN addition ($\sigma^2 = 10$)	0.1826	0.3553	0.5654	0.3311
	AWGN addition ($\sigma^2 = 20$)	0.2022	0.3348	0.5469	0.3817
Mobile	Identity	0.2607	0.2529	0.3145	1
	MPEG-2 Compression (64kbps)	0.2773	0.1777	0.2686	0.7617
	Spatial averaging (3×3)	0.2583	0.2591	0.3205	0.8839
	Spatial averaging (5×5)	0.2743	0.2575	0.3169	0.6317
	Brightness modification (+50%)	0.2891	0.2412	0.3125	0.9229
	Brightness modification (-50%)	0.2529	0.1973	0.3105	0.7520
	Contrast modification (HE)	0.2646	0.2461	0.2979	0.9121
	Frame dropping (50%, regular)	0.2637	0.2520	0.3145	0.9355
	Frame dropping (50%, random)	0.2221	0.2584	0.3051	0.8416
	AWGN addition ($\sigma^2 = 5$)	0.2651	0.2633	0.3251	0.7871
	AWGN addition ($\sigma^2 = 10$)	0.2676	0.2783	0.3232	0.6826
	AWGN addition ($\sigma^2 = 20$)	0.2750	0.2639	0.3194	0.6104

Table. 3.5: The similarity values for the original GOFs and the GOFs derived after the content-preserving operations when the $t5H$ band is used for representation

GOF	Content-preserving Operation	GOF			
		Coastguard	Container	Foreman	Mobile
Coastguard	Identity	1	0.3340	0.1914	0.2383
	MPEG-2 Compression (64kbps)	0.6426	0.3350	0.2070	0.2070
	Spatial averaging (3×3)	0.7959	0.3127	0.2093	0.2407
	Spatial averaging (5×5)	0.5820	0.2979	0.1865	0.1846
	Brightness modification (+50%)	0.8291	0.3301	0.1914	0.2383
	Brightness modification (-50%)	0.7744	0.2988	0.2168	0.2900
	Contrast modification (HE)	0.7334	0.3242	0.1982	0.2363
	Frame dropping (50%, regular)	0.8281	0.3477	0.2012	0.2520
	Frame dropping (50%, random)	0.8896	0.3585	0.2174	0.2909
	AWGN addition ($\sigma^2 = 5$)	0.7318	0.3363	0.2612	0.3488
	AWGN addition ($\sigma^2 = 10$)	0.5410	0.3740	0.2754	0.3613
	AWGN addition ($\sigma^2 = 20$)	0.5127	0.3874	0.2793	0.3874
Container	Identity	0.3340	1	0.2227	0.2813
	MPEG-2 Compression (64kbps)	0.2441	0.3857	0.2051	0.2344
	Spatial averaging (3×3)	0.3497	0.8250	0.2344	0.2819
	Spatial averaging (5×5)	0.3071	0.5794	0.2163	0.2944
	Brightness modification (+50%)	0.3171	0.7027	0.2959	0.2734
	Brightness modification (-50%)	0.3389	0.7891	0.2227	0.2813
	Contrast modification (HE)	0.3369	0.6045	0.2129	0.2744
	Frame dropping (50%, regular)	0.3350	0.7256	0.2197	0.2813
	Frame dropping (50%, random)	0.3745	0.7891	0.2212	0.2999
	AWGN addition ($\sigma^2 = 5$)	0.4816	0.4792	0.2904	0.2823
	AWGN addition ($\sigma^2 = 10$)	0.5410	0.3125	0.2998	0.2833
	AWGN addition ($\sigma^2 = 20$)	0.5773	0.2805	0.3612	0.3222
Foreman	Identity	0.1914	0.2227	1	0.1543
	MPEG-2 Compression (64kbps)	0.1914	0.2236	0.6455	0.1484
	Spatial averaging (3×3)	0.2275	0.1807	0.8486	0.1475
	Spatial averaging (5×5)	0.2166	0.2037	0.6402	0.1410
	Brightness modification (+50%)	0.2871	0.0625	0.3799	0.1583
	Brightness modification (-50%)	0.2021	0.2002	0.7891	0.1542
	Contrast modification (HE)	0.2344	0.2061	0.7715	0.1396
	Frame dropping (50%, regular)	0.1963	0.2246	0.8955	0.1563
	Frame dropping (50%, random)	0.1677	0.2165	0.9230	0.1621
	AWGN addition ($\sigma^2 = 5$)	0.2744	0.4372	0.7052	0.1836
	AWGN addition ($\sigma^2 = 10$)	0.2832	0.4783	0.5518	0.2021
	AWGN addition ($\sigma^2 = 20$)	0.2899	0.4726	0.5264	0.2103
Mobile	Identity	0.2383	0.2813	0.1543	1
	MPEG-2 Compression (64kbps)	0.2432	0.2617	0.1592	0.5771
	Spatial averaging (3×3)	0.2539	0.2651	0.1607	0.8357
	Spatial averaging (5×5)	0.2251	0.2508	0.1666	0.5893
	Brightness modification (+50%)	0.2373	0.2764	0.1572	0.9043
	Brightness modification (-50%)	0.2393	0.2773	0.1533	0.6465
	Contrast modification (HE)	0.2344	0.2773	0.1543	0.9424
	Frame dropping (50%, regular)	0.2354	0.2783	0.1514	0.8887
	Frame dropping (50%, random)	0.2220	0.2434	0.1481	0.8902
	AWGN addition ($\sigma^2 = 5$)	0.2518	0.2947	0.1480	0.7841
	AWGN addition ($\sigma^2 = 10$)	0.2666	0.2900	0.1592	0.6563
	AWGN addition ($\sigma^2 = 20$)	0.2631	0.2849	0.1718	0.5605

Table. 3.6: The similarity values for the original GOFs and the GOFs derived after the content-preserving operations when the $t4H$ band is used for representation

GOF	Content-preserving Operation	GOF			
		Coastguard	Container	Foreman	Mobile
Coastguard	Identity	1	0.4141	0.2354	0.4180
	MPEG-2 Compression (64kbps)	0.5635	0.4180	0.2148	0.4092
	Spatial averaging (3×3)	0.7129	0.3347	0.2128	0.4409
	Spatial averaging (5×5)	0.5029	0.4229	0.1914	0.4209
	Brightness modification (+50%)	0.8170	0.4141	0.2354	0.4180
	Brightness modification (-50%)	0.8555	0.4072	0.1934	0.4229
	Contrast modification (HE)	0.8057	0.4180	0.2217	0.4248
	Frame dropping (50%, regular)	0.8408	0.4277	0.2432	0.4268
	Frame dropping (50%, random)	0.8853	0.4344	0.2659	0.4166
	AWGN addition ($\sigma^2 = 5$)	0.9260	0.4362	0.2755	04307.
	AWGN addition ($\sigma^2 = 10$)	0.5527	0.4385	0.2891	0.4434
	AWGN addition ($\sigma^2 = 20$)	0.5068	0.4455	0.2971	0.4379
Container	Identity	0.4141	1	0.2070	0.3340
	MPEG-2 Compression (64kbps)	0.3691	0.3311	0.1982	0.3262
	Spatial averaging (3×3)	0.4210	0.7736	0.2194	0.3321
	Spatial averaging (5×5)	0.4181	0.5573	0.2017	0.3369
	Brightness modification (+50%)	0.4141	0.6375	0.2012	0.2197
	Brightness modification (-50%)	0.4141	0.8027	0.2100	0.3477
	Contrast modification (HE)	0.4111	0.6289	0.2031	0.3545
	Frame dropping (50%, regular)	0.4209	0.7100	0.2041	0.3516
	Frame dropping (50%, random)	0.4321	0.8076	0.2311	0.3759
	AWGN addition ($\sigma^2 = 5$)	0.5384	0.3201	0.3716	0.4193
	AWGN addition ($\sigma^2 = 10$)	0.5527	0.2875	0.4551	0.4443
	AWGN addition ($\sigma^2 = 20$)	0.5942	0.2678	0.4818	0.4601
Foreman	Identity	0.2354	0.2070	1	0.3320
	MPEG-2 Compression (64kbps)	0.2305	0.2080	0.5234	0.3262
	Spatial averaging (3×3)	0.2158	0.2275	0.7803	0.3320
	Spatial averaging (5×5)	0.2402	0.2019	0.5637	0.3374
	Brightness modification (+50%)	0.3564	0.1719	0.3018	0.2981
	Brightness modification (-50%)	0.2314	0.2246	0.7490	0.2073
	Contrast modification (HE)	0.2295	0.2285	0.7432	0.3184
	Frame dropping (50%, regular)	0.2305	0.2051	0.7764	0.3252
	Frame dropping (50%, random)	0.2390	0.2555	0.7984	0.3566
	AWGN addition ($\sigma^2 = 5$)	0.2868	0.4657	0.6331	0.3957
	AWGN addition ($\sigma^2 = 10$)	0.2930	0.5535	0.5293	0.3936
	AWGN addition ($\sigma^2 = 20$)	0.2983	0.5781	0.5078	0.3752
Mobile	Identity	0.4180	0.3340	0.3320	1
	MPEG-2 Compression (64kbps)	0.3838	0.3564	0.2734	0.5273
	Spatial averaging (3×3)	0.4093	0.3276	0.3108	0.7630
	Spatial averaging (5×5)	0.4188	0.3295	0.3362	0.5674
	Brightness modification (+50%)	0.4150	0.3330	0.3340	0.8516
	Brightness modification (-50%)	0.4023	0.3369	0.3340	0.6367
	Contrast modification (HE)	0.4180	0.3359	0.3340	0.8535
	Frame dropping (50%, regular)	0.4199	0.3340	0.3320	0.8740
	Frame dropping (50%, random)	0.4232	0.3170	0.3169	0.8059
	AWGN addition ($\sigma^2 = 5$)	0.4180	0.3874	0.3663	0.8014
	AWGN addition ($\sigma^2 = 10$)	0.4355	0.4414	0.3975	0.5811
	AWGN addition ($\sigma^2 = 20$)	0.4060	0.3981	0.3958	0.5615

(d) Experimental Verification of the Threshold

Before examining the overall performances of the various cases of representations, we determine a suitable value for the threshold T_2 experimentally and compare it with the value of 0.73 obtained through the statistical model. The 56 original GOFs from the 14 test videos are passed through the 11 content-preserving operations mentioned above. This derives 56 groups of GOFs, each group containing one original GOF and 11 content-preserved versions of the GOF. Except the GOFs derived after the frame-dropping operations, all other GOFs are temporally decomposed up to the fifth and fourth temporal levels separately. The temporal bands $t5L$, $t4L$, $t5H$ and $t4H$ are extracted. Equivalently, the GOFs derived after the frame-dropping operations are decomposed separately up to the fourth and third temporal levels, and the $t4L$, $t3L$, $t4H$ and $t3H$ bands are extracted.

We first find the histogram of the maximum Hamming distances for similar GOFs by separately using the four bands for representation. Consider the case of representation using the frame in the temporal band at the full level of temporal decomposition. Against each content-preserving operation, the perceptual blocks in the representative frame of each of the 56 GOFs derived after the content-preserving operation are compared with the corresponding perceptual block in the representative frame of the original GOF in the same group by using (3.6) and the maximum Hamming distance is computed. This derives 56 maximum Hamming distances per content-preserving operation and 616 maximum distances in total. When the other three cases of representations are also considered, altogether $4 \times 616 = 2464$ maximum Hamming distances are obtained. The histogram plot of the maximum Hamming distances at intervals of 8 units is shown in Figure 3.3(a).

Similarly, the histogram of the maximum Hamming distances for dissimilar GOFs is also computed by separately using the four bands for representation. Consider the case of representation using the frame in the temporal band at the full level of temporal decomposition. Against each of the identity and content-preserving operations, the perceptual blocks in the representative frame of each of the 56 GOFs derived after the operation are compared with the corresponding perceptual block in the representative frame of the original GOF in each of the other 55 groups by using (3.6). This derives $\frac{56\times55}{2} + 11 \times 56 \times 55 = 35420$ maximum Hamming Distance. For the four cases of representation, the number of maximum Hamming distance is $4 \times 35420 = 141680$. Figure 3.3(b) shows the histogram plot of the maximum Hamming distances at intervals of 8 units.

It can be observed in part (a) of Figure 3.3 that a very few numbers of maximum Hamming distances are greater than 160 for similar GOFs. For dissimilar GOFs, a few of the maximum

Hamming distances are smaller than 480 (part (b) in Figure 3.3). Hence, a value for the threshold T_1 may be considered between 160 and 480. The corresponding range for the threshold T_2 is [0.84 0.53]. This range includes the value of 0.73 computed from the statistical model for T_2. Hence, we consider $T_2 = 0.73$ during the experimentation in the following.

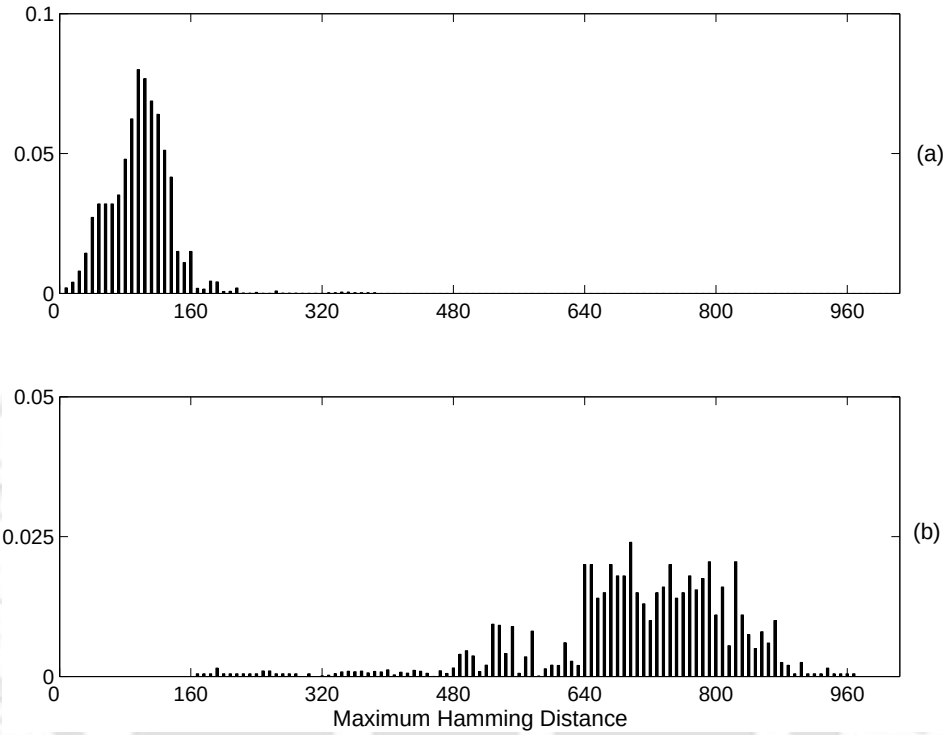


Fig. 3.3: The histograms of the maximum Hamming distances at intervals of 8 units when the temporal wavelet bands are used for representation: (a) similar GOFs and (b) dissimilar GOFs

(e) Overall Representation Performance

For computing the overall performances of the representative frames in all the four cases, each of the 56 original GOFs is passed through the 11 content-preserving operations mentioned above. Therefore, the 56 original GOFs derives 56 groups, where each group contains 12 similar GOFs including the original one. All GOFs are temporally decomposed up to the fifth and fourth levels separately except the GOFs derived after the frame-dropping operations. Equivalently, the GOFs derived after the frame-dropping operations are decomposed separately up to the temporal levels four and three.

It is expected that the representative frames for the original GOF and the representative frames for each of the other GOFs in a group result in similarity values greater than $T_2 = 0.73$. Consider

a case of representation among the four cases. Corresponding to each content-preserving operation, the similarity between the GOF derived after the content-preserving operation in each of the 56 groups and the original GOF in the same group is determined by using (3.8). This derives 56 similarity results corresponding to each content-preserving operation. The representation performance for content similarity R_1 against each content-preserving operation is computed by using (3.11). Similarly, the presentation performances against each content-preserving operation for the other cases of representation are computed.

Again, the representative frames for a pair of dissimilar GOFs are expected to result in a similarity value smaller than 0.73. For each case of representation, the similarities between each GOF in each group and the original GOFs in the other 55 groups are determined by using (3.8). This derives 3080 similarity results corresponding to a content-preserving operation. The representation performance for content dissimilarity R_2 against each content-preserving operation is computed by using (3.12). Finally, the values of the overall representation performance PI are computed by using (3.13). For each of the four cases of representation, $\%R_1$, $\%R_2$ and $\%PI$ against the content-preserving operations are shown in Table 3.7.

Table. 3.7: The representation performances of the $t5L$, $t4L$, $t5H$ and $t4H$ bands against the content-preserving operations: (A) identity, (B) MPEG-2 compression at the bit rate of 64kbps, (C) spatial averaging (1) 3×3 (2) 5×5, (D) brightness modification (1) +50% (2) -50%, (E) contrast modification (HE), (F) Frame dropping (1) 50% and regular (2) 50% and random (G) AWGN addition with variance (1) 5 (2) 10 (3) 20

	$t5L$			$t4L$			$t5H$			$t4H$		
	$\%R_1$	$\%R_2$	$\%PI$	$\%R_1$	$\%R_2$	$\%PI$	$\%R_1$	$\%R_2$	$\%PI$	$\%R_1$	$\%R_2$	$\%PI$
A	100	100	100	100	100	100	100	100	100	100	100	100
B	76.79	100	99.59	69.64	100	99.46	51.79	100	99.14	12.50	100	98.43
C1	96.43	100	99.94	96.43	100	99.94	94.64	100	99.90	91.07	100	99.84
C2	48.21	100	99.08	44.64	100	99.01	33.93	100	98.82	7.14	100	98.34
D1	85.71	100	99.74	89.29	100	99.81	75.00	100	99.55	75.00	100	99.55
D2	89.29	100	99.81	89.29	100	99.81	80.36	100	99.65	76.79	100	99.59
E	89.29	100	99.81	87.50	100	99.78	82.14	100	99.68	78.57	100	99.62
F1	96.43	100	99.94	96.43	100	99.94	94.64	100	99.90	83.93	100	99.71
F2	98.21	100	99.97	98.21	100	99.97	96.43	100	99.94	87.50	100	99.78
G1	100	100	100	100	100	100	85.71	100	99.74	76.79	100	99.59
G2	92.86	100	99.87	71.43	100	99.49	62.50	100	99.33	51.79	100	99.14
G3	25.00	100	98.66	19.64	100	98.57	8.93	100	98.37	0	100	98.21

It can be observed in Table 3.7 that the frames in the $t5L$ and $t4L$ bands perform better than the frames in $t5H$ and $t4H$ bands. Among the $t5L$ and $t4L$ bands, the single low-pass frame in

the $t5L$ band perform slightly better. Another advantage of using the $t5L$ band is the more concise representation with one frame in comparison to two frames in the case of the $t4L$ band. We therefore conclude that the single low-pass frame $f^{t\hat{u}L}$ at the full-level temporal decomposition can be used to represent the content of the GOF.

The low-pass frame $f^{t\hat{u}L}$ still contains the spatial redundancy. We concentrate in the following to explore the possibility of extracting a more concise representation for G_j by spatial decomposition on $f^{t\hat{u}L}$.

3.4 Content-Based Video Representation Using Spatio-Temporal Bands of 3D-DWT

It is observed in the previous section that the frame $f^{t\hat{u}L}$ in the low-pass band $t\hat{u}L$ at the full level of temporal decomposition of G_j contains sufficient information about the content of G_j and can be used for representing G_j. But, $f^{t\hat{u}L}$ cannot perform when G_j is spatially scaled. Using the t+2D variant of the 3D-DWT, the frames in the temporal wavelet bands of G_j are subjected to spatial decomposition. The different spatial bands derived from $f^{t\hat{u}L}$ may be explored for a concise representation of G_j.

3.4.1 Representative Frame from Spatio-Temporal Low-pass Band

Assume that the temporal wavelet bands of G_j are spatially decomposed up to an intermediate level v. When the frame $f^{t\hat{u}L}$ is spatially decomposed, one low-pass band at the highest level and three high-pass bands at each level of decomposition are obtained. Let us consider the high-pass bands $f^{t\hat{u}L-svLH}$, $f^{t\hat{u}L-svLH}$ and $f^{t\hat{u}L-svHH}$ at the v^{th} level of decomposition for representation. Exploiting the inherent spatial scalability of the 3D-DWT, these bands can be representations of the low-pass band $f^{t\hat{u}L-s(v-1)LL}$ and hence can support spatial scalability up to the $(v-1)^{\text{th}}$ level of spatial decomposition. For example, if spatial high-pass bands are considered at the second level of spatial decomposition of a GOF in the CIF format, it is possible to represent the GOF in the CIF and QCIF formats. If the spatial low-pass band at the third level is used, the representation is also possible when G_j is in the $\frac{1}{16}$CIF format. In general, $f^{t\hat{u}L-svLL}$ can support spatial scalability of the 3D-DWT up to the v^{th} level of spatial decomposition. Further, $f^{t\hat{u}L-svLL}$ being a thumbnail version of $f^{t\hat{u}L}$ [87], it should be the best concise representation for G_j. In [87], Panchanathan et al. use the low-pass DWT coefficients of video frames as features for detecting scene changes and observe

good efficiency in the video segmentation application. In the following, we investigate the possibility of using the $f^{t\hat{u}L-svLL}$ band as the representative frame f^{rep} for G_j. Note that this band is the spatio-temporal low-pass band $t\hat{u}L - svLL$ of G_j at the full level of temporal decomposition and at an intermediate level v of spatial decomposition. The two notations, $f^{t\hat{u}L-svLL}$ and $t\hat{u}L - svLL$, are interchangeably used in the remaining part of the thesis.

Representing the spatio-temporal low-pass content of G_j, the content of f^{rep} is expected to be robust against the content-preserving operations considered in the previous section in addition to the inherent spatio-temporal scalability of the 3D-DWT. The representation is expected to be more robust against noise because of the low-pass filtering involved in the extracting spatio-temporal bands. Algorithm 3.1 presents the steps for extracting the representative frame f^{rep} for a GOF.

Algorithm 3.1: Representative Frame Using the 3D-DWT of a GOF

Input
GOF G_j of size $N_1 \times N_2 \times N_3$
Level of the spatial decomposition v.

$\hat{u} = \lceil \log_2 N_3 \rceil$
Decompose G_j using the 3D-DWT up to the level $\hat{u}$ temporally and
up to the level v spatially
$f^{rep} = f^{t\hat{u}L-svLL}$

Output f^{rep}

3.4.2 Similarity Measure and Representation Performance

Similar to in the previous section, we propose to measure the similarity two representative frames at the perceptual block level. Binary versions of the representative frames are computed and are compared for determining the similarity between the corresponding GOFs.

(a) Perceptual Blocks and Binarisation

Consider two GOFs G_x and G_y with the representative frames f_x^{rep} and f_y^{rep} respectively. Similar to the cases of representation using the temporal wavelet bands, each representative frame is divided into M perceptual blocks of size $p \times p$, given by the indexed set $\{B^l | l = 1, 2, \cdots, M\}$. A perceptual

block of size $p \times p$ here represents a pixel volume of size $2^v p \times 2^v p \times N_3$ in the raw GOF. The size $p \times p$ should be appropriately chosen to the capture perceptual differences in the contents of distinct GOFs. For determining similarity between the two GOFs, the corresponding blocks are to be compared.

The wavelet coefficients in each perceptual block are binarised by means of a thresholding operation. As discussed in the previous section, here also thresholding about local mean is used. The values of 1 and 0 in the binarised data are not equally likely in this case. Our experimental observations on a number of binarised frames show this probability in the range [0.44 0.56]. We assume that each bit resulting from the mean thresholding takes values of 1 and 0 with the equal probability of 0.5. The binary frames $\hat{f}_x^{rep}$ and $\hat{f}_y^{rep}$ are derived by using the set of equations (3.4) and (3.5).

(b) Similarity Measure

The Hamming distance is used to compare the corresponding perceptual blocks in $\hat{f}_x^{rep}$ and $\hat{f}_y^{rep}$. It is computed according to (3.6). The similarity between G_x and G_y can be measured in terms of the $S(G_x, G_y)$ in (3.7). The two GOFs are concluded similar or dissimilar by applying (3.8).

(c) Representation Performance

Similar to the cases of representations using the temporal wavelet bands, the representation performance for content similarity and content dissimilarity are computed by using (3.11) and (3.12) respectively. The overall representation performance is measured in terms of the Performance Index given by (3.13).

3.4.3 Selection of the Threshold T_2

In Subsection 3.4.5, the spatio-temporal low-pass bands $t5L - s3LL$ and $t5L - s4LL$ of GOFs in the CIF format with 32 frames are considered for study of their performances as representative frames. It is observed experimentally there that 4×4 perceptual blocks in the $t5L - s3LL$ band and 2×2 perceptual blocks in the $t5L - s4LL$ band can fairly resolve the content similarities and dissimilarities in GOFs. In both the cases of representation, 99 perceptual blocks are derived from a representative frame.

In the case of representation by using the $t\hat{u}L - s3LL$ band, the probabilities of the maximum Hamming distances at $q = 0, 1, \cdots, 16$ are computed according to the model in (3.16). The plot of

these probabilities is shown in Figure 3.4(a). Using (3.17), the mean of the maximum Hamming distances is computed to be $\mu_\gamma = 12.846$. Therefore, the threshold T_1 is given by (3.18)

$$T_1 = \frac{\mu_\gamma}{2} = \frac{12.846}{2} = 6.423. \tag{3.23}$$

The threshold T_2 is obtained as (3.10):

$$T_2 = 1 - \frac{T_1}{p \times p} = 1 - \frac{6.423}{4 \times 4} = 0.599. \tag{3.24}$$

The probabilities of the maximum Hamming distances at $q = 0, 1, \cdots, 4$ are computed by using the model in (3.16) in the case of representation with the $t\hat{u}L - s4LL$ band. Figure 3.4(b) shows the plot of these probabilities. Using (3.17), the mean of the maximum Hamming distances is computed to be $\mu_\gamma = 3.998$ which derives $T_1 = 1.999$ and $T_2 = 0.5$. Table 3.8 shows the values for the thresholds T_1 and T_2 for the $t\hat{u}L - s3LL$ and $t\hat{u}L - s4LL$ bands.

Table. 3.8: The values obtained from the statistical model for T_1 and T_2 when GOFs in the CIF format are represented with the spatio-temporal low-pass bands

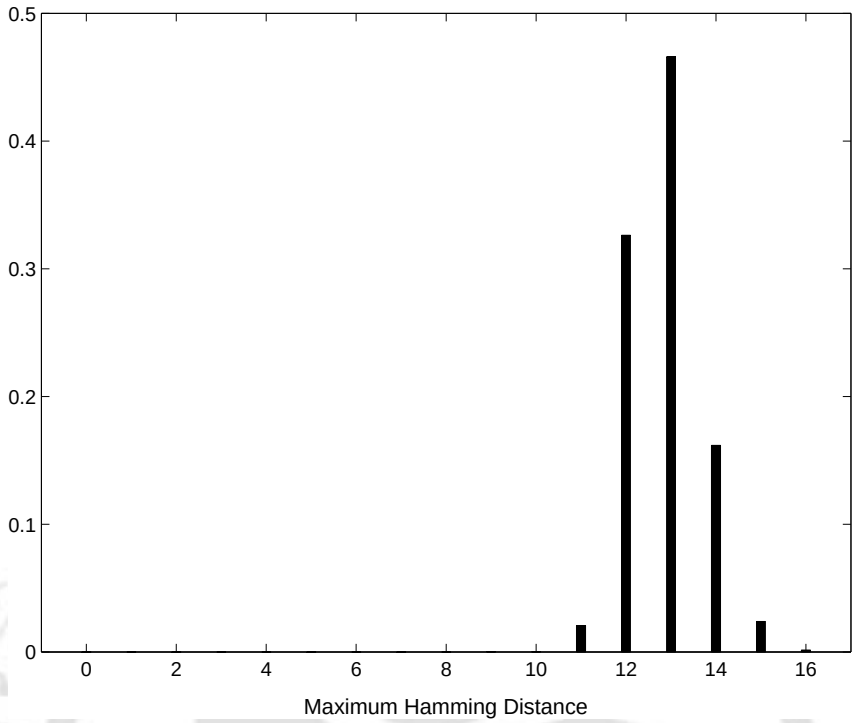
Spatio-temporal wavelet band	T_1	T_2
$t\hat{u}L - s3LL$	6.423	0.599
$t\hat{u}L - s4LL$	1.999	0.5

3.4.4 Spatio-Temporal Low-Pass Bands for Representation: an Analysis

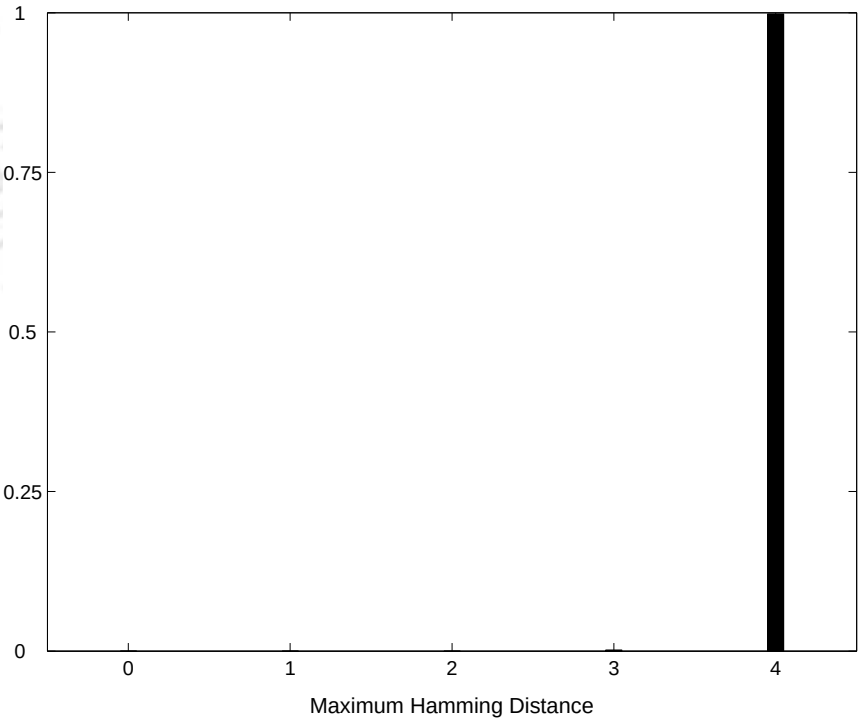
The spatio-temporal low-pass band of the 3D-DWT decomposition of a GOF considered for representation is at the full level of temporal decomposition and at an intermediate level of spatial decomposition. In the following, we discuss the strengths and weaknesses of using it as the representative frame for the GOF.

Sensitivity and size of perceptual block: As discussed later in Subsection 3.4.5, for a GOF in the CIF format with 32 frames, 4×4 perceptual blocks in the $t5L - s3LL$ band can fairly resolve the contents of GOFs. When the $t5L - s4LL$ band is used for representation, a suitable size is 2×2.

The perceptual blocks may be considered overlapping for detecting small differences in the contents of two representative frames that are present at the block boundaries distributing over multiple blocks.



(a) $p \times p = 4 \times 4$, $M = 99$



(b) $p \times p = 2 \times 2$, $M = 99$

Fig. 3.4: The probabilities of the maximum Hamming distances for distinct GOFs in the CIF format represented with the (a) $t\hat{u}L - s3LL$ band (b) $t\hat{u}L - s4LL$ band

Robustness against spatio-temporal scaling by wavelet transform: The content of the spatio-temporal low-pass band $t\hat{u}L - svLL$ of the GOF is naturally robust against the inherent temporal and spatial scalabilities of the 3D-DWT. It is robust up to the full-level of temporal and v^{th} level of spatial scalabilities of the 3D-DWT.

Robustness against MPEG compression: As the MPEG coders retain the low-frequency components of the visual data, the content of the $t\hat{u}L - svLL$ band is likely to be robust against the MPEG compression.

Robustness against spatial averaging: As the spatio-temporal low-pass content of the GOF is used for representation and the wavelet coefficients in each perceptual block are thresholded with respect to the mean of the coefficients in the block mean during the similarity verification, the contents of the $t\hat{u}L - svLL$ band is expected to be robust against the spatial-averaging operations on raw video frames.

Robustness against brightness and contrast modifications: Brightness modification is usually constant within a raw frame. The $t\hat{u}L - svLL$ band experiences a constant change due to a brightness modifications in frames. Therefore, brightness modifications should not affect the similarity between the representative frames of two GOFs. Again, because of high spatio-temporal correlation of video data, the contrast modification does not affect much the dynamic relationships of the coefficients in a block of a representative frame. Therefore the content of the $t\hat{u}L - svLL$ band is expected to be robust against contrast modifications also. But, the representation may fail when the brightness and contrast changes are beyond saturation.

Robustness against frame-rate changes: In a raw video, reduction in the frame rate is achieved by dropping frames either at a regular interval or at random. As the spatio-temporal low-pass content of a GOF is chosen for representing the GOF, it is likely to be robust against the frame-rate changes in the pixel domain. In the wavelet domain, the frame rate is reduced by truncating temporal high-pass bands selectively. Therefore, the $t\hat{u}L - svLL$ band remains available when the high-pass bands up to the levels $\hat{u}$ and $\hat{u} - 1$ are truncated.

3.4.5 Experimental Observations and Analysis II

In wavelet-based video coding, spatial decomposition is carried out usually up to the third or fourth level [52]. The low-pass bands at the full-level of temporal decomposition of each of the 56 original GOFs considered in Subsection 3.3.6 are spatially decomposed up to the third and fourth levels separately and the spatio-temporal low-pass bands $t5L-s3LL$ and $t5L-s4LL$ are extracted. We employ the Daubechies 9/7 biorthogonal wavelet filters [93] in the spatial decomposition for their energy-compaction capability [33]. The $t5L-s3LL$ and $t5L-s4LL$ bands of the GOFs derived after the content-preserving operations considered in Subsection 3.3.6 on the original GOFs are also similarly extracted.

(a) Selection of the Size of the Perceptual Blocks

When the $t5L-s3LL$ bands are considered for representing GOFs in the CIF format, it is experimentally observed that the perceptual blocks of size 4×4 contain significant perceptual information. They are also affected by the perceptual differences in contents of the distinct GOFs. This appears to be justified because the information contained by a 32×32 perceptual block in a $t5L$ band should be available in a $\frac{32}{2^3}\times\frac{32}{2^3}=4\times4$ perceptual block in the $t5L-s3LL$ band. Therefore, the $t5L-s3LL$ bands are divided into perceptual blocks of size 4×4. Following the same argument, the $t5L-s4LL$ bands are divided into perceptual blocks of size 2×2. The experimental results are presented below.

(b) Dissimilarity of GOF-wise Content

As in the cases of the temporal wavelet frames, the spatio-temporal low-pass bands of two neighbouring GOFs in a video should declare the GOFs as dissimilar. For demonstration, we again consider the four GOFs from each of the Coastguard, Container, Foreman and Mobile videos. The $t5L-s3LL$ and $t5L-s4LL$ bands of each GOF are extracted and the similarity values for the GOFs G_1 and G_2, G_2 and G_3 and G_3 and G_4 is computed by using (3.7). These results are shown in Table 3.9. When the $t5L-s3LL$ band is used for representation, the similarity values are below $T_2=0.599$ except for the GOFs G_1 and G_2 from the Container video. Using the $t5L-s4LL$ bands, similarity values equal to $T_2=0.5$ are also observed. These similarity values are obtained in the case of the GOFs from the slow Container video.

Table. 3.9: The similarity values for the adjacent GOFs from the Coastguard, Container, Foreman and Mobile videos by using the $t5L-s3LL$ and $t5L-s4LL$ bands for representation

GOF	$S(G_1,G_2)$		$S(G_2,G_3)$		$S(G_3,G_4)$	
	$t5L-s3LL$	$t5L-s4LL$	$t5L-s3LL$	$t5L-s4LL$	$t5L-s3LL$	$t5L-s4LL$
Coastguard	0.2500	0	0.1250	0	0.1250	0
Container	0.6250	0.5000	0.5000	0	0.5625	0.5000
Foreman	0.0625	0	0.0625	0	0.1875	0
Mobile	0.0625	0	0.1875	0	0.1250	0.2500

(c) Similarity of Contents under Content-Preserving Operations

For demonstrating the effects of the content-preserving operations on the $t5L-s3LL$ and $t5L-s4LL$ bands, the content-preserving operations considered in Subsection 3.3.6 are also considered here. The first GOFs from the Coastguard, Container, Foreman and Mobile videos are considered. The observed similarity values corresponding to the various content-preserving operations are presented in Table 3.10 and Table 3.11.

When the $t5L-s3LL$ band is used for representation, all the similarity values for the similar GOFs are greater than $T_2 = 0.599$ except for the contrast enhanced Container GOF. For dissimilar GOFs, these values are smaller than $T_2 = 0.599$. When the $t5L-s4LL$ band is used, the similarity values below $T_2 = 0.5$ are also observed for similar GOFs. The noisy Coastguard GOF (AWGN with $\sigma^2 = 20$) and the original Coastguard GOF have a similarity value of 0.2500. The noisy (AWGN with $\sigma^2 = 20$) and original Mobile GOFs also have a similarity value of 0.2500. The Mobile GOF results in a similarity value of 0.2500 under the frame-dropping operation.

(d) Experimental Verification of the Threshold

Before examining the overall representation performances of the representations using the $t5L-s3LL$ and $t5L-s4LL$ bands, we verify the model values of the thresholds in Table 3.8. The 56 original GOFs from the 14 test videos are passed through the 11 content-preserving operations mentioned above. This derives 56 groups of GOFs, each group containing one original GOF and 11 content-preserved versions of the GOF. Except the GOFs derived after the frame-dropping operations, all other GOFs are spatio-temporally decomposed to obtain the $t5L-s3LL$ and $t5L-s4LL$ bands. Equivalently, the frame-dropped GOFs are decomposed to derive the $t4L-s3LL$ and $t4L-s4LL$ bands.

Table. 3.10: The similarity values for the original GOFs and the GOFs derived after the content-preserving operations when the $t5L - s3LL$ band is used for representation

GOF	Operation	GOF			
		Coastguard	Container	Foreman	Mobile
Coastguard	Identity	1	0	0.0625	0.0625
	MPEG-2 Compression (64kbps)	0.7500	0	0.0625	0.0625
	Spatial averaging (3×3)	0.8125	0	0.0625	0.0625
	Spatial averaging (5×5)	0.7500	0	0.0625	0.0625
	Brightness modification (+50%)	1	0	0	0.0625
	Brightness modification (-50%)	1	0	0	0.0625
	Contrast modification (HE)	0.6875	0	0.1250	0.0625
	Frame dropping (50%, regular)	0.8750	0	0.0625	0.0625
	Frame dropping (50%, random)	0.7500	0	0.0625	0.0625
	AWGN addition ($\sigma^2 = 5$)	0.8125	0	0.0625	0.0625
	AWGN addition ($\sigma^2 = 10$)	0.7500	0	0.0625	0.0625
	AWGN addition ($\sigma^2 = 20$)	0.6250	0.0625	0.1250	0.0625
Container	Identity	0	1	0	0
	MPEG-2 Compression (64kbps)	0	0.8125	0	0
	Spatial averaging (3×3)	0	0.8750	0	0
	Spatial averaging (5×5)	0	0.8125	0.0625	0
	Brightness modification (+50%)	0	1	0	0
	Brightness modification (-50%)	0	1	0.1250	0.0625
	Contrast modification (HE)	0	0.5000	0.0625	0.0625
	Frame dropping (50%, regular)	0	0.9375	0	0
	Frame dropping (50%, random)	0	0.7500	0	0
	AWGN addition ($\sigma^2 = 5$)	0	0.6875	0	0
	AWGN addition ($\sigma^2 = 10$)	0	0.6875	0	0
	AWGN addition ($\sigma^2 = 20$)	0.0625	0.6250	0.1250	0.0625
Foreman	Identity	0.0625	0	1	0.0625
	MPEG-2 Compression (64kbps)	0.0625	0	0.8125	0.0625
	Spatial averaging (3×3)	0.0625	0	0.8750	0.0625
	Spatial averaging (5×5)	0.0625	0.0625	0.8125	0.0625
	Brightness modification (+50%)	0.0625	0.0625	0.6250	0.0625
	Brightness modification (-50%)	0.0625	0	0.8125	0.0625
	Contrast modification (HE)	0.0625	0	0.7500	0.0625
	Frame dropping (50%, regular)	0.0625	0	0.8750	0.0625
	Frame dropping (50%, random)	0.0625	0	0.8750	0.0625
	AWGN addition ($\sigma^2 = 5$)	0.0625	0	0.8750	0.0625
	AWGN addition ($\sigma^2 = 10$)	0.0625	0	0.8125	0.0625
	AWGN addition ($\sigma^2 = 20$)	0.0625	0.3125	0.7500	0.0625
Mobile	Identity	0.0625	0	0.0625	1
	MPEG-2 Compression (64kbps)	0.0625	0	0.0625	0.8750
	Spatial averaging (3×3)	0.0625	0	0	0.8750
	Spatial averaging (5×5)	0.0625	0	0	0.8125
	Brightness modification (+50%)	0.0625	0	0.0625	0.8125
	Brightness modification (-50%)	0.0625	0.0625	0.0625	0.6250
	Contrast modification (HE)	0.1250	0.0625	0.1250	0.8750
	Frame dropping (50%, regular)	0.0625	0	0.0625	0.9375
	Frame dropping (50%, random)	0.0625	0	0.0625	0.8125
	AWGN addition ($\sigma^2 = 5$)	0.0625	0	0.0625	0.8125
	AWGN addition ($\sigma^2 = 10$)	0.0625	0	0.0625	0.8750
	AWGN addition ($\sigma^2 = 20$)	0.0625	0	0.0625	0.8750

Table. 3.11: The similarity values for the original GOFs and the GOFs derived after the content-preserving operations when the $t5L - s4LL$ band is used for representation

GOF	Operation	GOF			
		Coastguard	Container	Foreman	Mobile
Coastguard	Identity	1	0	0	0
	MPEG-2 Compression (64kbps)	0.5000	0	0	0
	Spatial averaging (3×3)	0.7500	0	0	0
	Spatial averaging (5×5)	0.7500	0	0	0
	Brightness modification (+50%)	0.7500	0	0	0
	Brightness modification (-50%)	0.7500	0	0	0
	Contrast modification (HE)	0.5000	0	0	0
	Frame dropping (50%, regular)	0.7500	0	0	0
	Frame dropping (50%, random)	0.5000	0	0	0
	AWGN addition ($\sigma^2 = 5$)	0.7500	0	0	0
	AWGN addition ($\sigma^2 = 10$)	0.5000	0	0	0
	AWGN addition ($\sigma^2 = 20$)	0.2500	0	0	0
Container	Identity	0	1	0	0
	MPEG-2 Compression (64kbps)	0	0.5000	0	0
	Spatial averaging (3×3)	0	0.7500	0	0
	Spatial averaging (5×5)	0	0.7500	0	0
	Brightness modification (+50%)	0	0.7500	0	0
	Brightness modification (-50%)	0	0.7500	0	0
	Contrast modification (HE)	0	0.5000	0	0
	Frame dropping (50%, regular)	0	0.7500	0	0
	Frame dropping (50%, random)	0	0.5000	0	0
	AWGN addition ($\sigma^2 = 5$)	0	0.7500	0	0
	AWGN addition ($\sigma^2 = 10$)	0	0.7500	0	0
	AWGN addition ($\sigma^2 = 20$)	0	0.5000	0	0
Foreman	Identity	0	0	1	0
	MPEG-2 Compression (64kbps)	0	0	0.7500	0
	Spatial averaging (3×3)	0	0	0.7500	0
	Spatial averaging (5×5)	0	0	0.7500	0
	Brightness modification (+50%)	0	0	0.7500	0
	Brightness modification (-50%)	0	0	0.7500	0
	Contrast modification (HE)	0	0	0.5000	0
	Frame dropping (50%, regular)	0	0	0.7500	0
	Frame dropping (50%, random)	0	0	0.5000	0
	AWGN addition ($\sigma^2 = 5$)	0	0	0.7500	0
	AWGN addition ($\sigma^2 = 10$)	0	0	0.7500	0
	AWGN addition ($\sigma^2 = 20$)	0	0	0.5000	0
Mobile	Identity	0	0	0	1
	MPEG-2 Compression (64kbps)	0	0	0	0.7500
	Spatial averaging (3×3)	0	0	0	0.7500
	Spatial averaging (5×5)	0	0	0	0.7500
	Brightness modification (+50%)	0	0	0	0.7500
	Brightness modification (-50%)	0	0	0	0.7500
	Contrast modification (HE)	0	0	0	0.5000
	Frame dropping (50%, regular)	0	0	0	0.5000
	Frame dropping (50%, random)	0	0	0	0.2500
	AWGN addition ($\sigma^2 = 5$)	0	0	0	0.7500
	AWGN addition ($\sigma^2 = 10$)	0	0	0	0.5000
	AWGN addition ($\sigma^2 = 20$)	0	0	0	0.2500

The values for T_1 and T_2 for each of the $t5L-s3LL$ and $t5L-s4LL$ bands are verified separately by computing the histograms of the maximum Hamming distances for similar and dissimilar GOFs. In the case of representation using the $t5L-s3LL$ band, the suitable range for choosing a value for T_1 is found to be [4 12]. The corresponding range for T_2 is [0.25 0.75] which includes the model value of 0.599. Similarly, when the $t5L-s4LL$ band is considered for representation, the suitable range for considering a value for T_1 is found to be [1 3]. The corresponding range for T_2 in this case is [0.25 0.75]. This range includes the model value of 0.5. While examining the overall representation performances in the following, we consider the model values for T_2 in both the cases.

(e) Overall Representation Performance

Similar to in Subsection 3.3.6, the operation-wise representation performances for content similarity (R_1) and dissimilarity (R_2) are examined by using (3.11) and (3.12). The overall representation performances (PI) are computed by using (3.13). For each case of representation, $\%R_1$, $\%R_2$ and $\%PI$ against the content-preserving operations are shown in Table 3.12. It can be observed that the $t5L-s3LL$ band performs slightly better than the $t5L-s4LL$ band. Hence, for a GOF in the CIF format, the spatio-temporal band at the full level of temporal and third level of spatial decomposition may be used to represent the GOF.

Table. 3.12: The representation performances of the $t5L-s3LL$ and $t5L-s4LL$ bands against the content-preserving operations

Operation	$t5L-s3LL$			$t5L-s4LL$		
	$\%R_1$	$\%R_2$	$\%PI$	$\%R_1$	$\%R_2$	$\%PI$
Identity	100	99.48	99.49	100	99.42	99.43
MPEG-2 compression (64kbps)	100	99.48	99.49	100	99.38	99.39
Spatial averaging (3×3)	100	99.48	99.49	100	99.42	99.43
Spatial averaging (5×5)	100	99.35	99.36	100	99.22	99.23
Brightness modification (+50%)	100	99.74	99.74	100	99.74	99.74
Brightness modification (-50%)	100	99.74	99.74	100	99.68	99.69
Contrast modification (HE)	96.43	99.61	99.51	100	99.61	99.62
Frame dropping (50%, regular)	100	99.74	99.74	100	99.74	99.74
Frame dropping (50%, random)	100	99.74	99.74	94.64	99.81	99.71
AWGN addition ($\sigma^2=5$)	100	99.74	99.74	100	99.74	99.74
AWGN addition ($\sigma^2=10$)	100	99.74	99.74	98.21	99.74	99.71
AWGN addition ($\sigma^2=20$)	92.86	99.61	99.49	89.29	99.55	99.37

3.5 Discussion

It is observed in Subsection 3.3.6 that the frame in the $t\hat{u}L$ band of a GOF can perform as a representative frame for the GOF. Although the frames in the $t(\hat{u}-1)L$ band also perform almost equally, the consideration of the $t(\hat{u}-1)L$ band is requires to represent the GOF with two frames. The performances of the frames in the $t\hat{u}H$ and $t(\hat{u}-1)H$ bands are found to be inferior. It is observed in Subsection 3.4.5 that the spatio-temporal low-pass bands at the full level of temporal decomposition and at an intermediate level of spatial decomposition can be used to represent the GOF. For a GOF with 32 frames in the CIF format, the $t5L-s3LL$ band performs slightly better than the $t5L-s4LL$ band. The advantage of using the spatio-temporal low-pass band is the inherent robustness of its content against the spatio-temporal scalability features of 3D-DWT. Also, for wavelet-based scalably-coded video, the spatio-temporal low-pass bands of the GOFs can be extracted by partially decoding the coded bit-stream. This can be helpful in applications like real-time identification of video.

We employed the Haar wavelet filters in the temporal decomposition and the Daubechies 9/7 biorthogonal wavelet filters in the spatial decomposition. In the JPEG 2000 image coding standard, the 9/7 biorthogonal wavelet filters and the 5/3 LeGall wavelet filters are respectively used for lossy and lossless compression [94]. Similar restrictions in the case of WSVC framework are also expected.

From the observations made in this chapter, it should be possible to compute a hash of a video at the GOF level. The hash of a GOF may be extracted from the spatio-temporal band of the 3D-DWT decomposition of the GOF. We explore this possibility in the following chapter.

Chapter 4

Perceptual Hashing Using Block Averages in the 3D-DWT Band

It is observed in the previous chapter that the spatio-temporal low-pass band at the full level of temporal decomposition and at an intermediate level of spatial decomposition in the 3D-DWT domain can be used to represent the content of the GOF. This method of GOF representation can be used in the video identification application in a non-secured environment. It will be helpful if a hash of the GOF can be extracted from this band by taking into account the security aspect also.

There are a very few perceptual hash functions for video in the transform domain which consider the temporal information or the spatio-temporal information in totality during hash computation. For example, the hash function in [57] maps the temporal information in a video segment onto a TIRI and computes a hash of the segment from the content of the TIRI. The hash functions in [8], [56] consider the spatio-temporal information in video at a time. This chapter presents a new perceptual hash function in the 3D-DWT domain, which computes a hash of a GOF from the spatio-temporal low-pass content of the GOF. The robustness of the hash function is investigated against the scalability features of WSVC as well as against the common content-preserving operations on video. Experimental results are also presented to demonstrate the sensitivity of the hash function to content differences.

4.1 Desirable Properties of a Perceptual Hash Function

Consider a video V and a perceptual hash function h that derives a hash $h(V, K)$ of V by using a secret key K. The desirable properties of $h(V, K)$ has been discussed in Section 1.2.2. These

properties are summarised here for convenience.

i. $h(V, K)$ should be easily computable. Computational simplicity is particularly very important when $h(V, K)$ is used in real-time applications like identification of a video segment in a streaming video.

ii. $h(V, K)$ should be deterministic in nature. That is, it should be a deterministic function of V.

iii. It should be impossible to know V from its hash $h(V, K)$. Or, in other words, $h(V, K)$ should be one-way.

iv. The robustness property requires that the content-preserving operations like lossy compression, low-pass filtering, brightness and contrast modifications, etc. on V change $h(V, K)$ minimally.

v. The hash function should recognise the content differences in dissimilar videos. It should derive distinct hashes from distinct videos. This diffusion property ensures the fragility of $h(V, K)$ when the content of V is changed by an intruder.

vi. The detection of content differences in dissimilar videos alone may not be always sufficient in applications like content authentication, where the localisation of the differences may also be necessary. A hash function may be capable of resolving the differences in the temporal contents at the frame, segment, shot or video level and the differences in the spatial contents at the frame level down to the pixel level.

vii. The confusion property ensures the security of $h(V, K)$. It is expected in the security-related applications that when the secret key K is changed, the new hash should be drastically different from $h(V, K)$. Successful attacks on video by intruders become almost impossible when a secret key is used. Due to the secret key, a pirate also finds it very difficult to defy a copy detection system.

Apart from these properties, the real-time applicability of $h(V, K)$ may also be expected while identifying video segments in a real-time video. For that, the hash should be computed per segment of V and $h(V, K)$ should be obtained by concatenating the hashes of the segments.

4.2 Perceptual Hashing in the 3D-DWT based Scalable Coding Framework

A perceptual hash function in the scalable coding framework is advantageous if it is robust against the various scalability features: temporal, spatial and bit-rate. For the robustness against the scalability features of a SVC scheme, the hash function should compute a hash of a video from the base layer of the scalable bit-stream. As mentioned earlier, the perceptual hash function for image presented by Sun et al. in [54] is robust against the scalability features of the JPEG 2000 coding scheme. It computes a hash of an image from the compressed bit-stream at the output of the EBCOT block. The perceptual hashing of scalably-coded video is an unexplored area. A hash function, developed from the spatio-temporal low-pass band of the 3D-DWT decomposition of video may be robust against the 3D-DWT based scalabilities.

4.2.1 Features of 3D-DWT for Perceptual Hashing

We have established in the previous chapter that the spatio-temporal low-pass band at the full level of temporal and an intermediate level of spatial decomposition in the 3D-DWT domain can be used to represent the content a GOF. It is worthwhile now to explore if a hash of the GOF can be developed from this band. Noting that the GOF information is a part of the header bit-stream in WSVC schemes [22], such a hash function can be useful for the 3D-DWT coded scalable video and has the following advantages.

i. The spatio-temporal low-pass band corresponds to the base layer of a scalably-coded GOF. A hash function based on this band is automatically robust against the spatio-temporal scalability features of the coding scheme.

ii. The hash of the GOF inherits the spatio-temporal characteristic of the GOF. Any change in the characteristic should be reflected in the hash.

iii. The extraction of the hash from the spatio-temporal low-pass band also ensures the robustness of the hash against the transcoding operations in the 3D-DWT domain.

iv. The spatio-temporal low-pass band can be extracted by decoding the scalable bit-stream of the GOF partially. Hence, the hashing process is computationally simple.

v. As the frames in the GOF are highly correlated, the low-pass frame at the full-level of temporal

decomposition can summarise the GOF. But, the spatio-temporal low-pass band beyond a certain level of spatial decomposition lacks sufficient details and hence may not derive a workable hash.

vi. The spatio-temporal low-pass band may be divided into perceptual blocks [12] and a hash of the GOF may be obtained by combining the hashes of the blocks. This helps to resolve local differences in the contents of GOFs.

vii. A hash of the video may be obtained by concatenating the hashes of the GOFs.

Inspired by these features of the 3D-DWT, we concentrate in the following to design a perceptual hash function in the 3D-DWT domain such that it can also perform in the WSVC framework with robustness against the scalability features of SVC.

4.3 Perceptual Hashing Using Block Averages in the Spatio-temporal Low-pass Band

In the proposed solution, a hash of a GOF is computed from the spatio-temporal low-pass band $t\hat{u}L - svLL$. As mentioned earlier, $\hat{u}$ represents the full level of temporal decomposition and v represents an intermediate level of spatial decomposition of the GOF. We make the following assumptions:

i. The GOF structure of the video is available. A 3D-DWT based compression scheme compresses the video at the GOF level. Like the group-of-pictures (GOP) structure in the hybrid compression, the GOF structure is a part of the compressed bit-stream.

ii. The proposed hash function considers the GOFs in a video as independent entities and computes the hash of the video at the GOF level. It is therefore necessary to synchronise the GOFs in two videos during the comparison of their hashes.

In video coding using a WSVC scheme, the scalable bit-stream is arranged as follows [22]. The necessary header information, such as the fundamental characteristics of the video and the pointers to the beginning of each independent GOF bit-stream, is embedded at the beginning of the video bit-stream. This header is often called the *GOF header*. Hence, the bit-stream representing a GOF can be easily separated by using the information in the GOF header.

iii. The number of temporal and spatial decomposition levels are known and the necessary portion of the GOF bit-stream representing the required spatio-temporal band can be extracted. In video coding using a WSVC scheme, a GOF bit-stream carries headers that contain information about the temporal and spatial decomposition of the GOF. These headers also carry pointers to the beginning of the bit-stream portion at each level of temporal and spatial decomposition. Hence, the necessary portion of the bit-stream can be can be extracted.

iv. Unlike the DCT, the bases of the DWT are not unique. Therefore, the selection of the wavelet is also crucial for a DWT-based hash function. In the following, we employ the Haar wavelet bases [36], [49], [95] in the temporal decomposition and the Daubechies 9/7 biorthogonal wavelet bases in the spatial decomposition. The Haar filters are used along the temporal direction because they can handle smaller number of frames with less boundary problems. They also offer a good trade-off between the delay and the energy compaction [51]. Most works on wavelet-based video coding has used the Haar filters for temporal decomposition [45], [51], [95], [96]. The Daubechies 9/7 biorthogonal wavelet filters are used in the spatial decomposition for their energy-compaction capability [33]. These filters have been used by many 3D-DWT based video coding schemes [33], [97].

    A coding standard based on the DWT is expected to specify the wavelet. For example, the wavelet-based image coding standard JPEG 2000 restricts the Daubechies 9/7 biorthogonal [93] and 5/3 LeGall [98] bases corresponding to lossy and lossless compressions [94]. Similar restriction in the case of 3D-DWT based SVC schemes is also expected. When a hash of a GOF coded using a WSVC scheme is to be derived, the wavelets are to be defined by the compression standard.

The proposed methods for hash computation and hash comparison are described in the following.

4.3.1 Hash Computation

The content-based representation of a GOF by the spatio-temporal low-pass band is used to derive the hash. The basic idea here is to divide this representative frame into perceptual blocks and binarise the contents of these blocks by taking into account the security aspect. A simple block diagram for computing the hash of a GOF G is presented in Figure 4.1. The steps in the extraction of the hash are as follows.

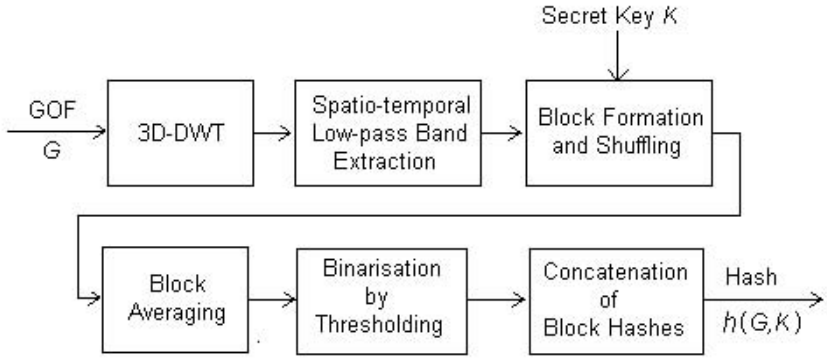


Fig. 4.1: Block diagram for hash computation using block averages in the 3D-DWT band

(a) Extraction of the Representative Band

Consider G to be of size $N_1 \times N_2 \times N_3$. Let the luminance component of G be decomposed fully along the temporal direction, i.e., up to the level $\hat{u} = \lceil \log_2 N_3 \rceil$. The resulting temporal low-pass band $t\hat{u}L$ is then decomposed spatially up to a level v to derive the spatio-temporal low-pass band $t\hat{u}L - svLL$. It was observed in the previous chapter that the $t5L - s3LL$ band of a 32-frame GOF in the CIF format contains substantial information for detecting content similarity / dissimilarity. It is used for representing the content of the GOF. The size of this band is 36×44. Alternatively, when a scalable bit-stream of a video V is available, the desired spatio-temporal low-pass bands of the GOFs may be obtained by partially decoding the bit-stream.

(b) Block formation and Shuffling

The $t\hat{u}L - svLL$ band is partitioned into M perceptual blocks of size $p \times p$. As discussed earlier, a perceptual block of size $p \times p$ represents a pixel volume of size $2^v p \times 2^v p \times N_3$ in G. The semantic information in two GOFs can be completely different even when one pair of the corresponding pixel volumes in the GOFs is content-wise different. Figure 3.1 showed the first frame in the GOF obtained by replacing one $32 \times 32 \times 32$ pixel volume in the first GOF from the Mobile video. To recognise such type of content differences in GOFs, the hash of each perceptual block in the $t\hat{u}L - svLL$ band is computed independently and the hash of G is obtained by combining the hashes of all the M perceptual blocks.

To introduce randomness during the hash computation, the perceptual blocks in the $t\hat{u}L - svLL$

band are randomly shuffled using a secret key K deriving the frame $\tilde{f}$. Let $\{B^l\,|l = 1, 2, \cdots, M\}$ represent the indexed set of the blocks after the random shuffling. As pointed out in Subsection 3.3.3, for robustness against the perceptually unimportant changes due to the content-preserving operations, the wavelet coefficients in each perceptual block can be thresholded.

(c) Binarisation by Thresholding

A thresholding process is used to binarise the perceptual blocks. Like in the earlier chapter, the local means of the perceptual blocks are considered as thresholds. For good diffusion property, a change in the perceptual content in a block should affect the hash bits of other perceptual blocks. For this purpose, the window W^l for computing the local mean is made to overlap with the neighbouring perceptual blocks as shown in Figure 4.2.

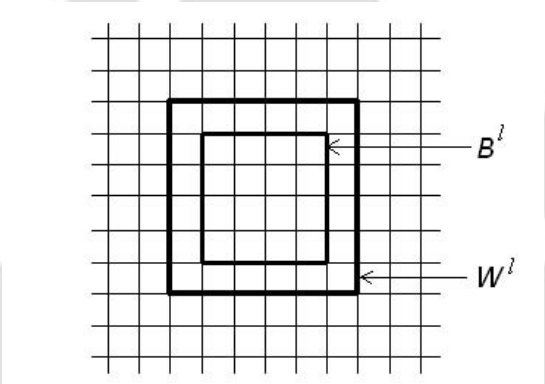


Fig. 4.2: The window of size 6×6 for computing the local mean of a 4×4 perceptual block

Suppose W^l is a block obtained by symmetrically augmenting the l^{th} perceptual block B^l to a size $w \times w > p \times p$. Then the wavelet coefficients in W^l also include the coefficients from the perceptual blocks surrounding B^l. A local mean μ^l for B^l is estimated by using

$$\mu^l = \frac{1}{w \times w} \sum_{(n_1,n_2)} \sum_{\in W^l} \tilde{f}(n_1, n_2). \tag{4.1}$$

When the key K is changed, the perceptual blocks surrounding B^l change and hence the local mean μ^l also changes. This imparts randomness to the local mean μ^l and hence improves the confusion property of the hash.

Each coefficient in B^l is thresholded with respect to the local mean μ^l. Let the wavelet coefficients $\tilde{f}(n_1, n_2)$, $(n_1, n_2) \in B^l$, in B^l be expressed as an indexed set $\{b^{l,q} | q = 1, 2, \cdots, p \times p\}$. The thresholding of the coefficients with respect to μ^l obtains $p \times p$ hash bits $a^{l,1}\ a^{l,2}\ \cdots\ a^{l,p\times p}$ of the block B^l according to:

$$a^{l,q} = \begin{cases} 1\ ; & \text{if } b^{l,q} \geq \mu^l \\ 0\ ; & \text{otherwise.} \end{cases} \tag{4.2}$$

The hash bits of each perceptual block constitute a hash for the perceptual block. The hash $h(G, K)$ of G is obtained by concatenating the hashes of the M blocks. Let $\|$ represent the concatenation operator [82]. Then $h(G, K)$ can be represented as

$$h(G, K) = a^{1,1}a^{1,2}\cdots a^{1,p\times p} \| a^{2,1}a^{2,2}\cdots a^{2,p\times p} \| \cdots \| a^{M,1}a^{M,2}\cdots a^{M,p\times p}. \tag{4.3}$$

Algorithm 4.1 presents the steps for hashing of a raw input GOF.

4.3.2 Hash Comparison

The proposed hash function considers the GOFs in a video as independent entities and computes the hash of the video at the GOF level. To verify the similarity of two videos, similarity verification of each pair of the corresponding GOFs in the two videos is required. Therefore, the proper synchronisation of the GOFs in the two videos is necessary for hash comparison. It is pointed out earlier that a scalable bit-stream of a video coded using a WSVC scheme includes the GOF information in a header. Hence, the GOFs in the two videos under comparison can be synchronised using the header information.

While computing the hash of a GOF using the proposed hash function, the hash of the individual perceptual block in the $t\hat{u}L - svLL$ band of the GOF is preserved. As discussed in the previous subsection, a perceptual block in the $t\hat{u}L - svLL$ band of the GOF corresponds to a pixel volume of size $2^v p \times 2^v p \times N_3$ in the GOF and the difference in the contents of two corresponding pixel volumes in two GOFs can potentially change the meaning of the GOFs. Hence, the hashes of two GOFs are compared at the perceptual-block level for resolving small / local differences in dissimilar GOFs. The two GOFs are concluded similar only when all the pairs of the corresponding perceptual blocks in their $t\hat{u}L - svLL$ bands are similar.

Algorithm 4.1: Hash Computation

Input
GOF: G of size $N_1 \times N_2 \times N_3$
Level of the spatial decomposition: v
Size of the perceptual block: $p \times p$
Size of the window for computing local mean: $w \times w$
Secret Key: K

$\hat{u} = \lceil \log_2 N_3 \rceil$
Decompose G using the 3D-DWT up to the level $\hat{u}$ temporally and up to the level v spatially
Extract the spatio-temporal low-pass band $t\hat{u}L - svLL$ of the decomposition
Divide $t\hat{u}L - svLL$ into M perceptual blocks of size $p \times p$ and randomly shuffle the perceptual blocks using K to derive the frame $\tilde{f}$. Let $\{B^l | l = 1, 2, \cdots, M\}$ represent the indexed set of blocks and $\{b^{l,q} | q = 1, 2, \cdots, p \times p\}$ represent the indexed set of the wavelet coefficients in the block B^l
For $l = 1, 2, \cdots, M$ **Do** //Loop on perceptual blocks
  Compute the local mean μ^l by using (4.1)
  For $q = 1, 2, \cdots, p \times p$ **Do** //Loop on perceptual block elements
    Compute $a^{l,q}$ by using (4.2)
  End
End

$h(G, K) = a^{1,1}a^{1,2}\cdots a^{1,p\times p} \| a^{2,1}a^{2,2}\cdots a^{2,p\times p} \| \cdots \| a^{M,1}a^{M,2}\cdots a^{M,p\times p}$

Output $h(G, K)$

Consider two GOFs G_x and G_y with hashes
$h(G_x, K) = a_x^{1,1}a_x^{1,2}\cdots a_x^{1,p\times p} \| a_x^{2,1}a_x^{2,2}\cdots a_x^{2,p\times p} \| \cdots \| a_x^{M,1}a_x^{M,2}\cdots a_x^{M,p\times p}$ and
$h(G_y, K) = a_y^{1,1}a_y^{1,2}\cdots a_y^{1,p\times p} \| a_y^{2,1}a_y^{2,2}\cdots a_y^{2,p\times p} \| \cdots \| a_y^{M,1}a_y^{M,2}\cdots a_y^{M,p\times p}$
respectively. We use the Hamming distance to compare the hashes of the corresponding perceptual blocks in $h(G_x, K)$ and $h(G_y, K)$. The Hamming distance between the hashes of the l^{th} perceptual blocks is obtained according to:

$$d_{x,y}^l = \sum_{q=1}^{p\times p} a_x^{l,q} \oplus a_y^{l,q}. \tag{4.4}$$

The two perceptual blocks are considered similar when $d^l_{x,y}$ is smaller than or equal to a suitably chosen threshold. Thus G_x and G_y are similar if

$$d^l_{x,y} \leq T_3 \quad ; \quad l = 1, 2, \cdots, M, \tag{4.5}$$

or equivalently

$$\max_{1 \leq l \leq M} \left(d^l_{x,y} \right) \leq T_3. \tag{4.6}$$

A normalised similarity measure $S(G_x, G_y)$ taking values between 0 and 1 is defined according to:

$$S(G_x, G_y) = 1 - \frac{1}{p \times p} \max_{1 \leq l \leq M} \left(d^l_{x,y} \right). \tag{4.7}$$

Therefore, G_x and G_y are similar if

$$S(G_x, G_y) \geq T_4, \tag{4.8}$$

where

$$T_4 = 1 - \frac{T_3}{p \times p}. \tag{4.9}$$

.

4.4 Salient Features of the Proposed Hash Function

The proposed hash function computes a hash of a GOF from the spatio-temporal low-pass band at the full level of temporal and an intermediate level of spatial wavelet decomposition of the GOF. The salient features of the proposed hash function are discussed in the following.

Sensitivity and perceptual block size: The sensitivity of a perceptual hash function depends on the smallest perceptual difference that the hash function can resolve. For the proposed hash function, the larger the perceptual blocks are, the lower is the sensitivity to tiny differences in contents. But,

when the perceptual blocks are very small, the hash function ceases to be robust against the content-preserving operations. It was observed in the previous chapter that for a GOF in the CIF format with 32 frames, 4×4 perceptual blocks in the $t5L - s3LL$ band fairly contain significant content information.

In addition, the scheme may fail to recognize small differences in the contents of two GOFs when they appear at the block boundaries distributing over multiple blocks. To overcome this limitation, perceptual blocks may be considered overlapping.

Size of the hash and hash space: Considering the $t5L - s3LL$ band of a GOF in the CIF format and the non-overlapping perceptual blocks, the proposed hash function derives a hash of size $M \times p \times p = 99 \times 4 \times 4 = 1584$ bits. As a compromise between the hash size and the sensitivity, the blocks may be considered half overlapping in the vertical and horizontal directions. The hash size is $3 \times 1584 = 4752$ bits in this case. These hash sizes are comparatively large.

When the perceptual blocks are non-overlapping, the cardinality of the cardinality of the hash space is $2^{1584} \approx 10^{477}$. The cardinality is $2^{4752} \approx 10^{1430}$ when the blocks are considered half overlapping. The hash spaces are very large in both the cases. As mentioned in Chapter 1, it is practically infeasible to find two videos which derive the same hash.

Key space: The space of the all possible keys that can be used with a hash function is called the key space of the hash function. The key space also determines the level of security provided by the hash function. For a hash function with a large key space, searching of the key used during hashing of a video becomes exhaustive. In the case of the proposed hash function, the secret key K is used to randomly shuffle the perceptual blocks. Hence, the number of possible keys is equal to the number of possible permutations. When the hash is extracted from the $t5L - s3LL$ band with a perceptual block size of 4×4, the 99 perceptual blocks can be shuffled in $99! \approx 10^{156}$ different ways. As expected, the key space of the hash function is very large.

Localization of content differences: The proposed hash function computes a hash of a GOF from the spatio-temporal low-pass band at the full level of temporal decomposition and at an intermediate level of spatial decomposition of the GOF. Hence, the differences in the temporal contents of videos can be localised at the GOF level. The hash function can localise differences in the spatial contents at the frame level.

Similar to the hash algorithm in [8], the proposed hash function can be used to identify a GOF

in a long video. In a 30 fps video, a GOF contains a clip of about 0.25, 0.5 or 1 second depending on 8, 16 or 32 frames in the GOF. This is important for resolving the perceptual content of a video in the temporal direction.

Computational Simplicity: As the spatio-temporal low-pass band can be obtained by partially decompressing the bit-stream of a WSVC-coded GOF, the proposed hash function is highly efficient when applied in the WSVC framework. Further, the hash computation from the spatio-temporal low-pass band also makes hash function efficient in the real-time application like the identification of video segments in streaming video.

Robustness against scalability features: As the hash of a GOF is computed from the spatio-temporal low-pass band of the GOF, it is naturally robust against the inherent spatio-temporal scalability features of the 3D-DWT and of a WSVC scheme. Again, due to the thresholding of the wavelet coefficients in each perceptual block with respect to a local mean for the block, the hash function is expected to be robust against the quantization operation and hence against the bit-rate scalability feature of the WSVC scheme.

Robustness against MPEG compression: Although the proposed hash function is designed to perform in the WSVC framework, its robustness against the MPEG compression is also important due to the popularity of the MPEG coding schemes. Video coders, including the MPEG coders, retain the low-frequency components of the visual data. As the proposed hash function computes a hash from the spatio-temporal low-pass content of video data, it is therefore likely to be robust against MPEG compression.

Robustness against spatial averaging: As mentioned earlier, the spatial averaging is used in noise smoothing, low-pass filtering and subsampling operations. Since the hash of a GOF is computed from the spatio-temporal low-pass band of the GOF, the proposed hash function is expected to be robust against the spatial averaging operations.

Robustness against brightness and contrast modifications: Brightness modification within a frame is usually constant. Hence, the coefficients of the spatio-temporal low-pass band of a GOF experience a constant change due to it brightness modification. Because of the thresholding of the wavelet coefficients with respect to the respective local mean, the hash of the GOF should not change due to the brightness modifications.

Again, because of high spatio-temporal correlation of video data, the contrast modification does not affect much the dynamic relationships of the coefficients in a perceptual block. Hence, the hash function is expected to be robust against the contrast modifications. However, when the brightness and contrast modifications are beyond saturation, the proposed hash function ceases to be robust due to the resulting uniform regions. But, then a GOF also loses its perceptual meaning.

Diffusion and confusion properties: The content of a perceptual block B^l affects local means for 9 perceptual blocks: the local mean for B^l and the local means for the 8 perceptual blocks surrounding B^l. When only one pair of the corresponding perceptual blocks of two dissimilar GOFs are dissimilar, the hash bits of maximum of 9 blocks may be different in the hashes of the GOFs. Hence, the proposed hash function has weak diffusion property. Because of the proposed similarity measure, the dissimilarity in one pair of corresponding perceptual blocks can also be detected.

On the other hand, when two different secret keys are used, the locations of two perceptual blocks during the random shuffling may be different in the worst case. In that case, the local means of maximum of 18 blocks may be affected. Thus a maximum of about 20% of hash bits may change. Therefore, the confusion property of the hash function is also weak.

4.5 Experimental Observations and Analysis

To study the performance of the proposed hash function, the luminance components of the 14 test videos used in the last chapter are considered for experimentation. The videos are Akiyo, Antibes, Bike, Cheer, Coastguard, Container, Football, Foreman, Garden, Mobile, Mosaic, News, Stefan and Tempete. These videos are in the CIF format with a frame rate of 30 fps. Four GOFs, each of 32 frames, are used from each of the videos.

It was observed in the earlier chapter that the spatio-temporal low-pass band $t5L - s3LL$ of a GOF can be used to represent the content of the GOF. The hashes of the GOFs are computed from their $t5L - s3LL$ bands. We apply the Haar filters to decompose the GOFs along the temporal direction and the Daubechies 9/7 biorthogonal wavelet filters for the spatial decomposition. The 3D-DWT coefficients are quantized using 8 bits to retain only the integer parts of the coefficients. As pointed out earlier, perceptual blocks of size 4×4 contain significant perceptual information and are also affected by the perceptual differences in the contents of dissimilar GOFs. Hence, the perceptual block size in the experiments here is 4×4. The $t5L - s3LL$ band results in 99 perceptual

blocks of size 4×4. We consider the non-overlapping perceptual blocks here. A symmetric window of size 6×6 is considered for computing the local means. Therefore, a GOF derives a hash of 1584 bits.

The proposed hash function is tested for robustness against the scalability features of the 3D-DWT / WSVC and against content-preserving operations. The sensitivity of the hash function to the content differences is also examined. The following operations are considered on the original GOFs.

(a) Wavelet-based Scalability

For testing the robustness of the hash function against the spatio-temporal scalabilities of the 3D-DWT / WSVC schemes, the following operations are performed on each wavelet-decomposed original GOF.

i. Temporal resolution reduction: The GOF can be temporarily scaled by dropping the high-pass frames above certain levels. Since the hash is extracted from the highest level of temporal decomposition, it is naturally robust against the temporal scaling in the 3D-DWT domain.

ii. Spatial resolution reduction: Similar to the temporal scaling in the 3D-DWT domain, spatial high-pass bands above certain levels can be dropped for the spatial scaling of the GOF. As the hash of the GOF is extracted from the $t5L - s3LL$ band, the high-pass bands at the first, second and third levels of spatial decomposition can be truncated without affecting the hash. In other words, the hash function is naturally robust against the spatial scaling of the GOF in the 3D-DWT domain up to the spatial dimension of the $\frac{1}{64}$CIF format.

iii. Bit-rate resolution reduction: Quantizers can be used to mimic the bit-rate resolution reduction scenarios [99]. Four quantizers, 8-bit, 7-bit, 6-bit and 5-bit, are separately used to quantize the wavelet coefficients in the $t5L - s3LL$ band of each GOF to reduce the quality or bit-rate resolution.

Due to the natural robustness of the proposed hash function against the temporal and spatial resolution reductions, the experimental results for the quantization operations are presented here. The hashes of the GOFs are computed after quantizing the wavelet coefficients in the $t5L - s3LL$ bands by applying the 8-bit, 7-bit, 6-bit and 5-bit quantizers separately.

(b) Content-Preserving Operations

The following content-preserving operations are considered. The GOFs derived after each operation are wavelet decomposed and the 3D-DWT coefficients in the $t5L - s3LL$ bands are quantized using a 8-bit quantizer. The hashes of the GOFs are then computed from the quantized $t5L - s3LL$ bands. The following operations are considered.

i. MPEG compression: To examine the effect of the lossy compression on the contents of the representative frames, the GOFs are compressed using the MPEG-2 coder at the bit-rate of 64kbps and decompressed. The decompressed GOFs are considered for experimentation.

ii. Spatial averaging: For studying the robustness of the hash function against the spatial-averaging operation, averaging masks of sizes 3×3 and 5×5 are applied to the frames of the GOFs. This derives two GOFs per original GOF.

iii. Brightness modification: As mentioned above, the brightness modification within a frame is usually constant. The intensity of each frame in the GOFs is increased/decreased by 50% of the original frame intensity.

iv. Contrast modification: For enhancing the contrast of the GOFs, the histogram equalisation is applied on the frames of the GOFs.

iv. Noise addition: To test the robustness of the hash function against the AWGN, the GOFs are corrupted with zero-mean AWGN with variance of 10 / 20.

As a result of the quantization and content-preserving operations, we obtain a group of 12 hashes of similar GOFs for each of the 56 original GOFs. The 12 hashes include the hash of the original GOF, the three hashes computed after quantizing the $t5L - s3LL$ band of the original GOF with each of the three quantizers and the eight hashes computed from the GOFs obtained after the content-preserving operations on the original GOF.

(c) Content Differences

A hash function should be sensitive to the global and local differences in the contents of GOFs. For examining the sensitivity of the proposed hash function, content differences at the block, frame and GOF levels are considered. We modify the original GOFs to have GOFs with content differences at the block and frame levels.

i. Block-level content difference: The proposed similarity measure declares two GOFs dissimilar even when one pair of corresponding perceptual blocks in their spatio-temporal bands are detected dissimilar. One perceptual block of size 4×4 occupies approximately 1% of the area of the $t5L - s3LL$ band. To study the performance of the hash function against the content differences at the block level, a block is modified in the same position in each frame of the GOFs. Different sizes of blocks starting with the size of the frame are considered for experimentation. It is observed that the hash function can detect content modifications up to a block size of 1% of the frame size when the perceptual blocks are half overlapping in the horizontal and vertical directions. When the perceptual blocks are non-overlapping, the hash function detects content modification up to a block size of about 5% of the frame size. This is due to the fact that the replacing blocks in the frames may belong to multiple pixel volumes corresponding to multiple perceptual blocks. With the non-overlapping perceptual blocks, the case of 5% block size is reported here.

   In each original GOF, another original GOF chosen at random is inserted in such a way that an inserted frame occupies 5% of the area of a frame. The 56 original GOFs results in 56 modified GOFs, where the corresponding GOFs have differences in their contents at the block level.

ii. Frame-level content difference: To be perceptible to the viewers, at least a few successive frames in two GOFs should be different [83]. Different numbers of consecutive frames, starting from 32 frames per GOF are considered for replacement. It is observed that the hash function can detect frame replacement up to 8 consecutive frames (spreading over a duration of approximately 250 msec) per GOF. The performance of the hash function against the replacement of 8 consecutive frames per GOF is reported here.

   To derive GOFs with frame-level differences in their contents, the last eight frames in each original GOF are replaced with equal number of frames from another original GOF chosen at random. This derives another set of 56 modified GOFs.

iii. GOF-level content difference: To study the performance of the hash function in distinguishing content differences at the GOF level, the pairs of distinct GOFs are considered. For the 56 distinct original GOFs, $\frac{56 \times 55}{2} = 1540$ pairs of GOFs with dissimilar contents are possible.

In all the above cases, the similarity of two GOFs is determined by using (4.7) and (4.8).

4.5.1 Demonstrating the Working of the Hash Function

The working of the proposed hashing solution is first demonstrated with the first GOF from the Mobile video. The three quantizers, 7-bit, 6-bit and 5-bit, are used to quantize the wavelet coefficients in the $t5L - s3LL$ band of the GOF separately and the hash is computed after each quantization operation. The original GOF is subjected to the other content-preserving operations considered above and the hashes of the resulting GOFs are computed after quantizing the $t5L - s3LL$ bands using a 8-bit quantizer. The observed similarity values by using (4.7) for different cases are presented in Table 4.1. The observations are as follows.

Table. 4.1: The robustness of the hash function against the quantization and content-preserving operations: demonstration with the first GOF from the Mobile video

Operation	Similarity value, S
Quantization: 8-bit	1
Quantization: 7-bit	0.8750
Quantization: 6-bit	0.7500
Quantization: 5-bit	0.8125
MPEG-2 compression	0.8750
Spatial averaging: 3×3 mask	0.9375
Spatial averaging: 5×5 mask	0.7500
Brightness modification: +50%	0.9375
Brightness modification: -50%	0.8125
Contrast modification (HE)	0.8125
AWGN addition: $\sigma^2 = 10$	0.8125
AWGN addition: $\sigma^2 = 20$	0.7500

i. Quantization: The similarity value between the $t5L - s3LL$ band of the original GOF and the $t5L - s3LL$ bands after quantizing by using 8-bit, 7-bit, 6-bit and 5-bit quantizers are found to be 1, 0.8750, 0.7500 and 0.8125 respectively. These similarity values suggest the robustness of the hash function against the quantization and hence against the bit-rate resolution reduction in the WSVC framework.

ii. MPEG compression: For MPEG-2 compression at 64kbps, the observed similarity value is 0.8750. This indicates the robustness of the proposed hash function against the MPEG-2 compression.

iii. Spatial averaging: When the saptial-averaging masks of sizes 3×3 and 5×5 are applied on each frame of the GOF, the observed similarity values are 0.9375 and 0.7500 respectively. These

values suggest the resistance of the hash function against the spatial-averaging operation.

iv. Brightness modification: When the brightness of each frame in the GOF is increased by 50% of the brightness of the frame, the similarity value is 0.9375. For the case of brightness decrement by 50%, the observed similarity value is 0.8125. The two similarity values suggest robustness of the hash function against brightness modification.

v. Contrast modification: The contrast of each frame in the GOF is enhanced by histogram equalisation. For the contrast enhanced GOF, the similarity value is 0.8125, which indicates the resilience of the hash function against the contrast modification.

vi. AWGN addition: For the cases of AWGN addition, the observed similarity values are 0.8125 ($\sigma^2 = 10$) and 0.7500 ($\sigma^2 = 20$). These values also suggest the robustness of the hash function against the AWGN.

A hash function should be sensitive to the global as well as local differences in the contents of GOFs. For demonstrating the sensitivity, the following observations are presented.

i. Block-level content difference: For testing the sensitivity to content difference at the block level, the month-mark '1' in the frames of the GOF is changed with '2'. The first frames from the original and modified GOFs are shown here again in Figure 4.3. The observed similarity value of 0.1825 (Table 4.2) suggests the sensitivity of the proposed hash function to the block-level differences in contents.

ii. Frame-level content difference: The last eight frames of the GOFare replaced with the corresponding frames in the first GOF from the Foreman video. The observed similarity value of 0.3750 (Table 4.2) suggests the sensitivity of the hash function to the content differences at the frame level.

iii. GOF-level content difference: The first GOFs from the Mobile and Foreman videos results in a similarity value of 0.1250 (Table 4.2). It indicates the sensitivity of the hash function to the content differences at the GOF level.

4.5.2 Average Performances

The average performances of the proposed hash function against the quantization and content-preserving operations and against the various types of content differences are presented in the following.

(a) (b)

Fig. 4.3: The first frames from the (a) original Mobile GOF (b) modified Mobile GOF.

Table. 4.2: The sensitivity of the hash function to the content differences at various levels: demonstration with the first GOF from the Mobile video

Content Difference	Similarity value, S
Block level	0.1825
Frame level	0.3750
GOF level	0.1250

(a) and (b) Quantization and Content-Preserving Operations

We first examine the average performances of the hash function in terms of the average similarity values against the quantization and other content-preserving operations. The 56 groups of hashes, each containing 12 hashes of similar GOFs, are considered in the experimentation. For example, in the case of the contrast modification operation, the similarity value for each of the 56 GOFs derived after the contrast enhancement and the corresponding original GOF is computed by using (4.7). The average similarity value against the contrast modification operation is thus average of 56 similarity values. For robustness, this value should be close to one. The average similarity value against each of the 12 operations is computed. The white bars in Figure 4.4 show the average similarity value obtained against the identity, quantization and content-preserving operations. Note that the identity operation refers to the quantization of the wavelet coefficients by 8 bits. The observed average similarity values are close to one which indicates the robustness of the hash function against these operations.

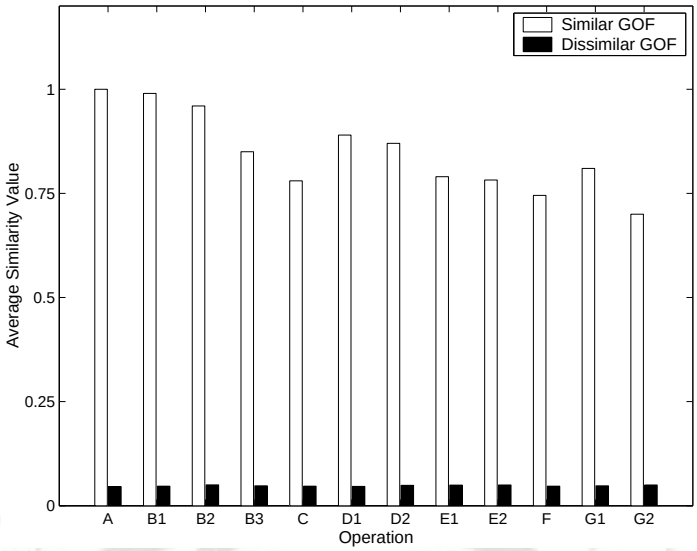


Fig. 4.4: The average performances of the hash function in terms of the average similarity values against the operations: (A) identity, (B) quantization (1) 7-bit (2) 6-bit (3) 5-bit, (C) MPEG-2 compression at the bit rate of 64kbps, (D) spatial averaging (1) 3×3 (2) 5×5, (E) brightness modification (1) +50% (2) -50%, (F) contrast modification (HE), (G) AWGN addition with variance (1) 10 (2) 20

(c) Content Differences

For studying the average performances against the differences in the contents of GOFs at the GOF level, the similarity values are computed for dissimilar GOF pairs. In the case of the identity operation, the similarity value between each original GOF and each of the other 55 original GOFs is computed by using (4.7). The average similarity value against the operation is thus the average of the $\frac{56\times55}{2} = 1540$ similarity values. Against any of the quantization and content-preserving operations, the average similarity value is the average of $56 \times 55 = 3080$ similarity values. The average similarity values against the various operations are also shown in Figure 4.4 with the black bars. These values are very close to zero. This indicates that the proposed hash function recognises the content differences in dissimilar GOFs very well. Moreover, it can be observed in the figure that the minimum of the average similarity values is 0.71 in the case of similar GOFs. For dissimilar GOFs, the maximum average similarity value is 0.054. The large difference between the two indicates the goodness of the proposed hash function.

As mentioned above, we also present results here for the differences in the contents of GOFs at the block level (the case of 5% block size) and frame level (the case of replacement of 8 consecutive frames per GOF). In the case of block-level content difference, the similarity value for each of the 56

original GOFs and the corresponding modified GOF with block-level content difference is computed by using (4.7) and the average similarity value is calculated. Similarly, the similarity value for each original GOF and the corresponding modified GOF with frame-level content difference is computed. The average of the 56 similarity values is calculated. The two average similarity values are presented in Figure 4.5. The average similarity value for the GOF-level content difference against the identity operation in Figure 4.4 is also presented in Figure 4.5 for comparison. These values are close to zero indicating that the proposed hash function is sensitive enough to detect content differences at the block, frame and GOF levels.

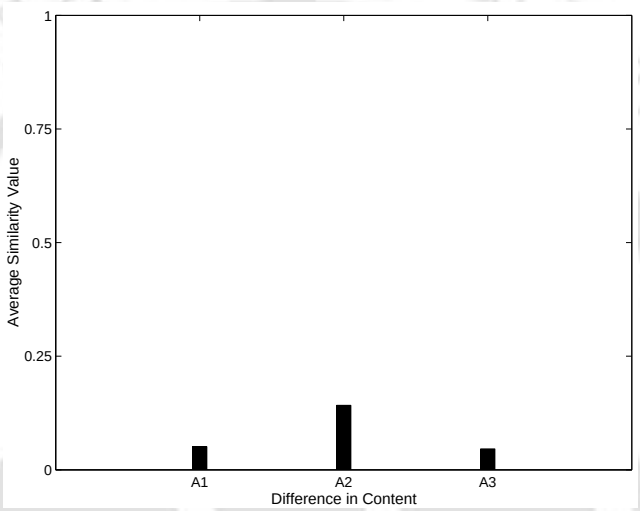


Fig. 4.5: The average performances of the hash function in terms of the average similarity values against the content differences at the: (A1) block level (for the case of 5% block size) (A2) frame level (for the case of replacement of 8 consecutive frames per GOF) (A3) GOF level

4.5.3 Experimental Verification of the Threshold

The performance of the hash function is presented with histograms showing the normalised frequencies of the similarity values. The 56 groups of hashes, where each group contains the hashes of 12 perceptually similar GOFs, are considered. Within each group, the hash of the original GOF and the hash of each of the other 11 perceptually similar GOFs are compared by using (4.4). The corresponding similarity values are computed by using (4.7). This results in 616 similarity values from the 56 groups. The histogram of the normalised frequencies of these similarity values is shown with the light bars in Figure 4.6. Further, computing the similarity values for the all-possible pairs of the dissimilar original GOFs, 1540 values are obtained. The histogram of the normalised frequencies of the similarity values in this case is shown with the dark bars in Figure 4.6. The two histograms

suggest a value for the threshold T_4 between 0.250 and 0.625 for distinguishing perceptually similar and dissimilar GOFs. This range of values for T_4 also include the value 0.599 for the $t5L - s3LL$ band obtained in the previous chapter from the statistical model.

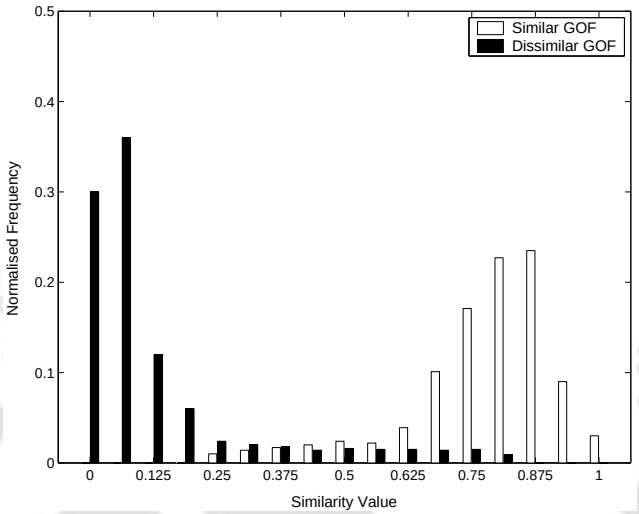


Fig. 4.6: The histograms of the normalised frequencies of the similarity values

4.5.4 Verification Performance

The verification performance of the hash function can be measured in terms of the rates of false rejection and false acceptance. The false rejection rate (FRR) is the rate of erroneously declaring similar GOFs as dissimilar and the false acceptance rate (FAR) is the rate of incorrectly declaring dissimilar GOFs as similar [100]. In Chapter 3, we used C_1 and C_2 respectively to represent the number of content-wise similar GOF pairs declared as similar and the number of similar GOF pairs under test. Hence, the FRR is given by

$$FRR = 1 - \frac{C_1}{C_2}. \tag{4.10}$$

Similarly, C_3 and C_4 were respectively used to represent the number of content-wise dissimilar GOF pairs declared as dissimilar and the number of dissimilar GOF pairs under test. The FAR is thus obtained according to:

$$FAR = 1 - \frac{C_3}{C_4}. \tag{4.11}$$

For an ideal hash function, the FRR and FAR values are zero. In the case of practical hash functions, non-zero rates of false rejection and false acceptance may be observed. But, these rates should be very close to zero. During the analysis of the verification performance of the proposed scheme, similarity between two GOFs is concluded by using (4.7) and (4.8) with $T_4 = 0.599$.

The 56 groups of hashes, where each group contains the hashes of 12 perceptually similar GOFs, are considered. Within each group, the hash of the original GOF and the hash of each of the other 11 perceptually similar GOFs are compared. The similarity of each pair of GOFs is determined. The FRR rates against the identity, quantization and content-preserving operations are computed and are presented in Figure 4.7 with the light bars. The similarities of the all-possible pairs of the dissimilar original GOFs are also determined. The FAR rates against the identity, quantization and content-preserving operations are also presented in Figure 4.7 with the dark bars. Nonzero FRR rates are observed in the cases of the contrast modification and AWGN with variance of 20. The FRR of 3.5% in the case of the contrast modification operation may be due to the changes in the spatio-temporal low-pass content of the GOFs introduced by the histogram equalisation. The FRR in the case of the AWGN addition with $\sigma^2 = 20$ is found to be 7.1% indicating that AWGN with high variance affects the performance of the hash function. Against the all operations, the FAR rates are non-zero but well below 1%. The non-zeros FAR rates are due to erroneously declaring some of the adjacent distinct GOFs from the slow videos (like the Container and News videos) as similar.

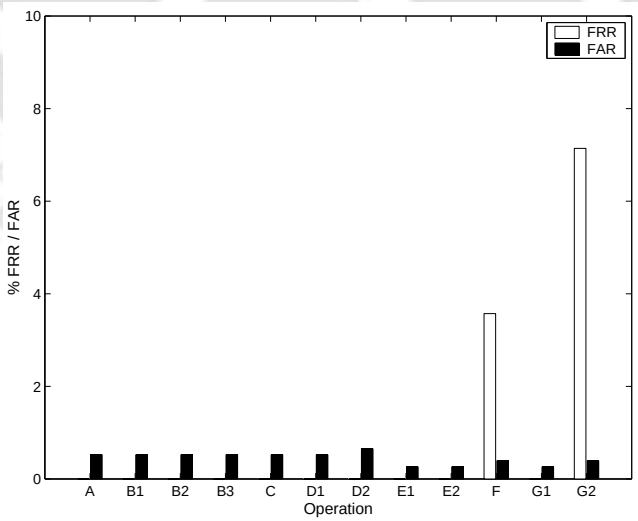


Fig. 4.7: The FRR and FAR rates against the operations: (A) identity, (B) quantization (1) 7-bit (2) 6-bit (3) 5-bit, (C) MPEG-2 compression at the bit rate of 64kbps, (D) spatial averaging (1) 3×3 (2) 5×5, (E) brightness modification (1) +50% (2) -50%, (F) contrast modification (HE), (G) AWGN addition with variance (1) 10 (2) 20

4.6 Discussion

In this chapter, a perceptual hash function for video in the 3D-DWT domain has been presented. It computes the hash of a GOF from the spatio-temporal low-pass band at the full level of temporal decomposition and at an intermediate level of spatial decomposition of the GOF. The hash function can perform in the WSVC framework. During the hash comparison, the GOFs in two scalably-coded videos can be synchronised using the header information in the bit-streams. The hash and key spaces for the hash function are very large. Although the hash function has weak diffusion and confusion properties, the observed good performance is due to the novel similarity measure used for hash verification. The robustness of the hash function against the scalability features of 3D-DWT / WVSC and the other content-preserving operations has been examined. The sensitivity of the hash function to the content differences at the block, frame and GOF levels has been studied. The observed FRR and FAR rates are nominal. As the hashes are computed from the spatio-temporal low-pass contents, the hash function in a few cases fails to resolve contents in distinct GOFs from very slow videos like the Container and News videos. Due to operating at the GOF level, the hash function may be also used for identifying small video segments in a secured database or in a broadcast in real-time.

The large hash size is a limitation of the proposed hash function. Further, the diffusion and confusion properties of the hash function are also weak. Designing of perceptual hash functions deriving shorter hashes and having good diffusion and confusion properties is considered in the next chapter.

Chapter 5

Perceptual Hashing Using Cumulative Block Averages in the 3D-DWT Band

The hash function proposed in the previous chapter has two major drawbacks: (i) large hash size and (ii) weak diffusion and confusion properties. It is observed that for a GOF in the CIF format, it derives a hash of size 1584 bits with non-overlapping perceptual blocks and of size 4752 when the perceptual blocks are half overlapping in the both horizontal and vertical directions. These hash sizes are larger than one kilo-bits. Due to the large hash sizes, a larger bandwidth is required to transmit the hash in the video authentication application, which may not be always acceptable. Moreover, the time complexity of hash comparison is also high because of the large hash size and the similarity measure.

This chapter presents a perceptual hash function which also computes a hash from the spatio-temporal low-pass band at the full-level of temporal decomposition and at an intermediate level of spatial decomposition of the GOF. This hash function derives hashes of a smaller size in comparison to the hash size in the previous chapter. The robustness of the hash function is studied against the scalability features of the 3D-DWT / WSVC and against the common content-preserving operations on video. Experimental results are presented to demonstrate the sensitivity of the hash function to content differences. The confusion and diffusion properties of the hash function are also studied.

5.1 Perceptual Hashing Using Cumulative Block Averages in Spatio-temporal Low-pass Band

It is observed in Chapter 3 that the spatio-temporal low-pass band at the full-level of temporal and an intermediate level of spatial decomposition of a GOF contains sufficient information about the content of the GOF for representation. A hash of the GOF was extracted from this band in the previous chapter. Considering the attractive features of the 3D-DWT for video hashing discussed in Subsection 4.2.1, we design in this section a perceptual hash function for deriving a compact hash of the GOF from the aforesaid band. Similar to the hash function presented in the previous chapter, resolving of local differences in the contents of distinct GOFs is also considered here. The assumptions made in the previous chapter about the GOFs and the wavelet bases are also valid here.

5.1.1 Hash Computation

Consider a GOF G of size $N_1 \times N_2 \times N_3$. It is decomposed fully along the temporal direction and then decomposed spatially up to a level v to derive the spatio-temporal low-pass band $t\hat{u}L - svLL$. The following points are considered for extracting a hash of the GOF.

i. To cancel the effect of brightness modifications, the mean μ of the spatio-temporal low-pass band $t\hat{u}L - svLL$ is subtracted from the wavelet coefficients in the band.

ii. For capturing the local content of G, the mean-subtracted band is divided into M perceptual blocks of size $p \times p$ like in the previous chapter.

iii. To protect the hash, the perceptual blocks are randomly shuffled by using a secret key K. Let the indexed set of the blocks after the random shuffling be $\{B^l | l = 1, 2, \cdots, M\}$.

iv. Let $\{\mu^l | l = 1, 2, \cdots, M\}$ be the set of corresponding mean values of the perceptual blocks. These mean values are representatives of the local contents of $t\hat{u}L - svLL$. Thus, a hash of G may be computed from the mean values.

v. For good diffusion property, a large number of hash bits of two distinct GOFs should be different even for a local difference in their contents. Thus, it is desirable that the mean of a perceptual block affects multiple hash bits. This suggests for derivation of each hash bit by involving multiple means. We propose to consider the *forward cumulative averages* and the *backward*

*cumulative average*s of the block means for extracting the perceptual hash. The forward and backward cumulative averages and their binarisation are discussed below.

(a) Forward and Backward Cumulative Averages

Consider the indexed set of block means $\{\mu^l | l = 1, 2, \cdots, M\}$. The forward cumulative average is given by [101]

$$\mu_f^l = \frac{1}{l}\sum_{i=1}^{l}\mu^i \quad ; \quad l = 1, 2, \cdots, M. \tag{5.1}$$

The forward cumulative average can be recursively updated as

$$\mu_f^l = \frac{l-1}{l}\mu_f^{l-1} + \frac{1}{l}\mu^l \quad ; \quad l = 1, 2, \cdots, M. \tag{5.2}$$

Note that the forward cumulative average at any location l contains information about the mean at l and all the locations preceding l. In other words, the average μ^l at location l affects the cumulative averages at the locations $l, l+1, \cdots, M$. Thus, for larger l, the information about μ^l is contained in fewer cumulative averages. For example, the information about μ^{M-1} is contained in μ_f^{M-1} and μ_f^M. The information about μ^M is contained in μ_f^M only. Hence, the forward cumulative averages alone can not derive a good hash.

To overcome the limitation of using the forward cumulative averages alone in this application, we also consider the cumulative averages of $\{\mu^l | l = 1, 2, \cdots, M\}$ computed along the backward direction. These averages are to be called the *backward cumulative average*s and are obtained according to:

$$\mu_b^l = \frac{1}{M-l+1}\sum_{i=l}^{M}\mu^i \quad ; \quad l = 1, 2, \cdots, M. \tag{5.3}$$

The backward cumulative average can be recursively updated by using

$$\mu_b^l = \frac{M-l}{M-l+1}\mu_b^{l+1} + \frac{1}{M-l+1}\mu^l \quad ; \quad l = M, M-1, \cdots, 1. \tag{5.4}$$

The recursive computation simplifies the computations of the cumulative averages and it can be helpful in the case of real-time applications of the hash function. From the above discussion, there

are altogether $2M$ cumulative averages for M perceptual blocks out of which $M+1$ cumulative averages will be affected even when one perceptual block at any location is modified.

(b) Binarisation of the Forward and Backward Cumulative Averages

The forward and backward cumulative averages are binarised to get the hash. The binarisation process makes the hash robust against the content-preserving operations on G.

The binarisation of the cumulative averages can be done by a suitable thresholding operation. It is pointed out in Chapter 3, the advantage of the median thresholding is that it results in equal number of 1's and 0's in the binarised data and we can assign a probability of 0.5 for each bit to take the value of 1 or 0. As the volume of data to be ordered for finding the median is not very high, the median-based thresholding is employed here. The forward and backward cumulative averages are considered as two independent indexed set for binarisation.

Let med_f and med_b represent the median of the forward cumulative averages $\mu_f^l | l = 1, 2, \cdots, M$ and the median of the backward cumulative averages $\mu_b^l | l = 1, 2, \cdots, M$ respectively. The thresholding operation derives a hash $h(G, K) = a^1\, a^2\, \cdots\, a^{2M}$ of G according to:
For $l = 1, 2, \cdots, M$,

$$a^l = \begin{cases} 1\,; & \text{if} \quad \mu_f^l \geq med_f \\ 0\,; & \text{otherwise} \end{cases} \tag{5.5}$$

and

$$a^{M+l} = \begin{cases} 1\,; & \text{if} \quad \mu_b^l \geq med_b \\ 0\,; & \text{otherwise.} \end{cases} \tag{5.6}$$

The block diagram for computing $h(G, K)$ is presented in Figure 5.1. The first three blocks are identical for the algorithm developed in the last chapter. Algorithm 5.1 presents the steps of the hashing strategy for a raw input GOF.

5.1.2 Hash Comparison

The proposed hash function considers the GOFs in a video as independent entities and computes the hash of the video at the GOF level. To verify the similarity of two videos, similarity verification of each pair of the corresponding GOFs in the two videos is required. Therefore, the proper synchronisation of the GOFs in the two videos is necessary for hash comparison. It is pointed out earlier that a scalable bit-stream of a video coded using a WSVC scheme includes the GOF information in a

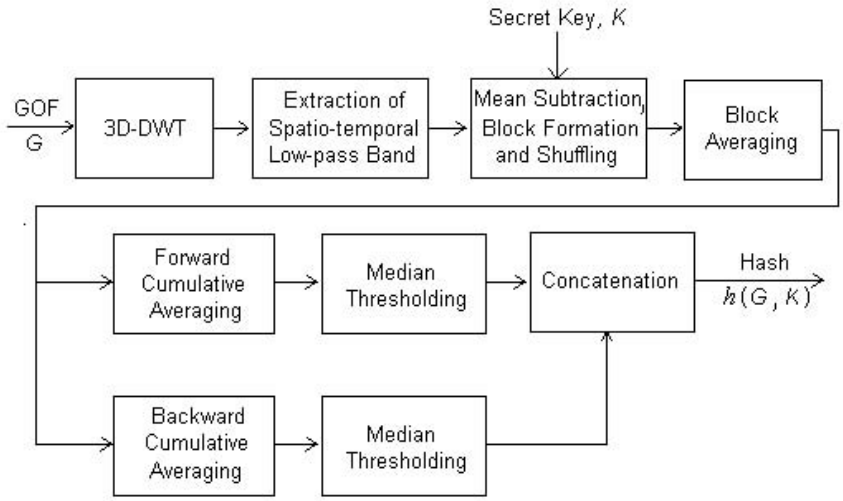


Fig. 5.1: Block diagram for hash computation using cumulative block averages in the 3D-DWT band

header. Hence, the GOFs in the two videos under comparison can be synchronised by using the header information.

Consider two GOFs G_x and G_y with the hashes $h(G_x, K) = a_x^1 a_x^2 \cdots a_x^{2M}$ and $h(G_y, K) = a_y^1 a_y^2 \cdots a_y^{2M}$ respectively. We use the Hamming distance to compare the hashes according to:

$$d(h(G_x, K), h(G_y, K)) = \sum_{i=1}^{2M} a_x^i \oplus a_y^i. \tag{5.7}$$

The Hamming distance can be normalised and a similarity value $S(G_x, G_y)$ in the range $[0 \quad 1]$ indicating the similarity between the two GOFs is computed by using

$$S(G_x, G_y) = 1 - \frac{1}{2M} d(h(G_x, K), h(G_y, K)). \tag{5.8}$$

The two GOFs G_x and G_y are declared similar when

$$d\left(h(G_x, k), h(G_y, K)\right) \leq T_5 \tag{5.9}$$

or equivalently

$$S(G_x, G_y) \geq T_6. \tag{5.10}$$

Algorithm 5.1: Hash Computation

Input
GOF: G of size $N_1 \times N_2 \times N_3$
Level of the spatial decomposition: v
Size of the perceptual block: $p \times p$
Secret Key: K

$\hat{u} = \lceil \log_2 N_3 \rceil$
Decompose G using the 3D-DWT up to the level $\hat{u}$ temporally and up to the level v spatially
Extract the spatio-temporal low-pass band $t\hat{u}L - svLL$ of the decomposition
Compute the mean μ of the $t\hat{u}L - svLL$ band
Subtract μ from each wavelet coefficient in the $t\hat{u}L - svLL$ band
Divide the mean-subtracted $t\hat{u}L - svLL$ band into M perceptual blocks of size $p \times p$ and randomly shuffle the blocks using the secret key K to obtain the indexed set of blocks $\{B^l | l = 1, 2, \cdots, M\}$
Compute the indexed set of means $\{\mu^l | l = 1, 2, \cdots, M\}$, where μ^l is the mean of the wavelet coefficients in B^l
For $l = 1, 2, \cdots, M$ **Do** //Loop on perceptual block means
  Compute the forward cumulative average μ_f^l using (5.2)
  Compute the backward cumulative average μ_b^l using (5.4)
End
Compute the median med_f of the indexed set $\{\mu_f^l | l = 1, 2, \cdots, M\}$
Compute the median med_b of the indexed set $\{\mu_b^l | l = 1, 2, \cdots, M\}$
Compute $h(G, K) = a^1\, a^2 \cdots a^{2M}$ using (5.5) and (5.6)

Output $h(G, K)$

The thresholds T_5 and T_6 are chosen suitably and are related by

$$T_6 = 1 - \frac{T_5}{2M}. \tag{5.11}$$

(a) Selection of the Threshold T_6

Due to the use of the median-based quantization in the hash computation, the Hamming distance between the hashes of two distinct GOFs is always even with equal numbers of 1's and 0's in disagreement. Hence, we can write $d = 2d'$, where $d' = 0, 1, 2, \cdots, M$ represent the number of 1's or 0's in disagreement.

Let the probability mass function (PMF) of d' be $P_{d'}(l), l = 0, 1, 2, \cdots, M$. We assume that the

hash bits are independent. Since each bit in a hash takes a value of 1 or 0 with equal probability, $P_{d'}(l)$ is binomial with mean $\mu_{d'} = \frac{M}{2}$ and variance $\sigma^2_{d'} = \frac{M}{4}$ [56]. Therefore, the PMF of the Hamming distance d is obtained as $P_d(2l) = P_{d'}(l)$ with mean $\mu_d = M$ and variance $\sigma^2_d = M$ [56]. Assume that two similar GOFs have ideally zero Hamming distance. The threshold T_5 may be set at

$$T_5 = \frac{0 + \mu_d}{2} = \frac{M}{2}. \tag{5.12}$$

For $M = 99$, an approximate plot of the probability mass function P_d is shown in Fig. 5.2. In this case, $T_5 = 49.5$ and T_6 is computed by using (5.11) as

$$T_6 = 1 - \frac{T_5}{2M} = 1 - \frac{49.5}{198} = 0.75. \tag{5.13}$$

As $\mu_d - T_5 \approx 5\sqrt{\sigma^2_d}$, the probability of erroneously accepting dissimilar GOFs as similar is very small (approximately of the order of 10^{-7}). In Subsection 5.3.3, a value for T_6 will be evaluated based on the experimental observations and compared with the value predicted by the model.

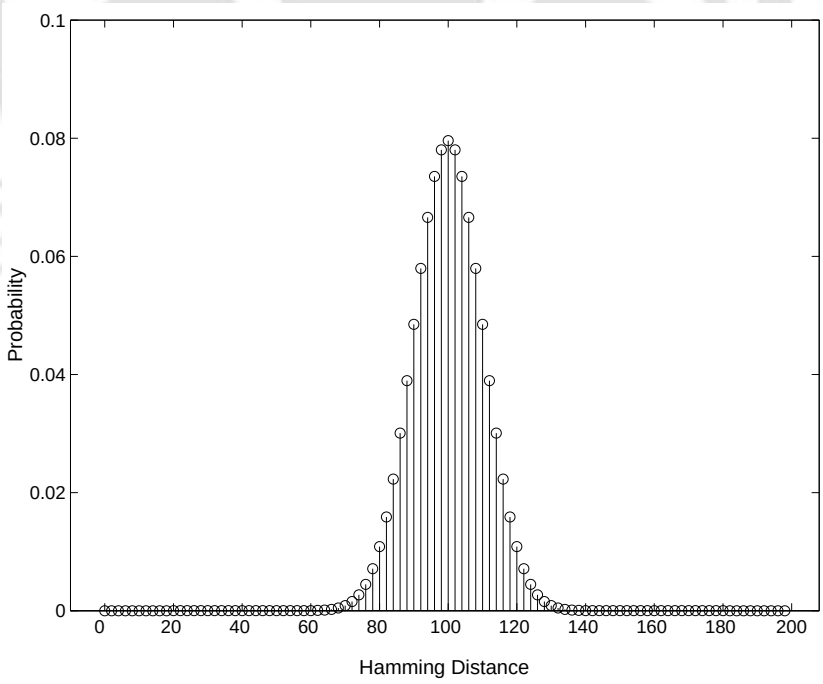


Fig. 5.2: The PMF of the Hamming distances between hashes of distinct GOFs ($M = 99$)

5.2 Salient Features of the Proposed Hash Function

The proposed hash function computes the hash of a GOF from the spatio-temporal low-pass band of the 3D-DWT decomposition of the GOF. The full level of temporal decomposition and an intermediate level of spatial decomposition are considered. In the following, we discuss some salient features of the proposed hash function.

Sensitivity and perceptual block size: As mentioned earlier, the sensitivity of a video hash function depends on the smallest difference in the contents of two video sequences that the function can detect. For the proposed function, the smaller the perceptual blocks are, the higher is the sensitivity to the differences in the contents of two GOFs. The smaller the block size results in a larger number of blocks in the spatio-temporal band of the GOF and a longer hash. Hence, one has to compromise among the sensitivity of the function, the hash length and the possible number of hashes. For a 32-frame GOF in the CIF format, it was observed that 4×4 blocks in the spatio-temporal band $t5L - s3LL$ fairly contain significant content differences. It follows from the subsequent discussion that the number of possible hashes is sufficiently large with this block size and the length of the hash is moderate.

Similar to the hash function presented in the previous chapter, this hash function may also fail to discriminate small differences in the contents of two GOFs when they appear at the block boundaries distributed over multiple blocks. The perceptual blocks may be considered overlapping to enhance the sensitivity of the hash function to such differences.

Size of the hash and hash space: When the $t5L-s3LL$ band is divided into 4×4 non-overlapping perceptual blocks, the proposed hash function derives a hash of 198 bits. As a compromise between the sensitivity and the hash size, the perceptual blocks may be considered half overlapping in the vertical and horizontal directions. In this case, the hash size increases to 594 bits. A longer hash is the price for enhancing the sensitivity of the function.

When the perceptual blocks are non-overlapping, the cardinality of the hash space is $2^{198} \approx 10^{59}$. The cardinality is $2^{594} \approx 10^{178}$ if the blocks are considered half overlapping. As mentioned in Chapter 1, it is practically infeasible in both the cases to find two GOFs that derive the same hash.

Key space: In the proposed hash function, the secret key K is used to randomly shuffle the perceptual blocks. When the hash is extracted from the $t5L - s3LL$ band with a perceptual block

size of 4×4, the 99 perceptual blocks can be shuffled in $99! \approx 10^{156}$ different ways. When the hash function is used in a video authentication system, it will not be easy for an attacker to successfully attack video due to the large hash and key spaces.

Localization of content differences: The proposed hash function computes a hash from the spatio-temporal low-pass band at the full level of temporal and an intermediate level of spatial decomposition of the GOF. Hence, the differences in the temporal contents of videos can be localised at the GOF level. The hash function can localise differences in the spatial contents at the frame level.

Computational Simplicity: As the spatio-temporal low-pass band can be obtained by partially decompressing the scalable bit-stream of the GOF, the proposed hash function is highly efficient when applied in the WSVC framework. Further, the hash computation from the spatio-temporal low-pass band also makes a real-time application of the hash function possible.

Robustness against scalability features: As the hash is computed from the spatio-temporal low-pass band of the GOF, it is naturally robust against the spatio-temporal scalability features of 3D-DWT / WSVC. Due to the thresholding of the cumulative block averages about their median, the hash function is expected to be robust against the quantization operation and hence against the bit-rate scalability.

Robustness against MPEG compression: Although the proposed hash function is ideally suitable for the video sequences coded using WSVC schemes, due to the widespread popularity of the MPEG video coders, it should be able to handle the MPEG-coded video sequences also. As mentioned earlier, the MPEG coders retain the low-frequency components of the video data. As the hash is computed from the spatio-temporal low-pass content of the GOF, the hash function is likely to be robust against MPEG compression.

Robustness against spatial averaging: As the hash function computes the hash from the spatio-temporal low-pass content of the GOF, it is expected to be robust against the spatial averaging operations.

Robustness against brightness modifications: The brightness modification within a frame is usually constant. As the mean of the wavelet coefficients in the spatio-temporal low-pass band is

subtracted from each coefficient before the hash computation, the hashes of a GOF before and after the brightness modification are same. However, when the brightness is modified beyond saturation, the hash function loses the robustness due to the resulting in uniform regions. But, then a GOF also loses its perceptual meaning.

Diffusion and confusion properties: Suppose that one pair of corresponding perceptual blocks in two spatio-temporal low-pass bands is different. This will change $M + 1$ cumulative averages out of the total of $2M$ forward and backward cumulative averages. Similarly, more than half of the $2M$ cumulative averages change when the secret key is changed. Hence, the proposed hash function is expected to have good diffusion and confusion properties.

5.3 Experimental Observations and Analysis

We consider the luminance components of the 56 original GOFs from the 14 test videos in the CIF format used in the previous two chapters for experimentation. These videos are Akiyo, Antibes, Bike, Cheer, Coastguard, Container, Football, Foreman, Garden, Mobile, Mosaic, News, Stefan and Tempete. Four GOFs, each of 32 frames, are used from each video.

The hashes of the GOFs are computed from their $t5L - s3LL$ bands obtained by applying the Haar filters along the temporal direction and the Daubechies 9-7 biorthogonal wavelet filters along the spatial directions of the GOF. Accordingly, the perceptual block size in the following experiments is also 4×4. The perceptual-blocks are considered non-overlapping. As mentioned earlier, a GOF derives a hash of 198 bits. The following operations on the GOFs are considered.

The hash function is tested for robustness against the scalability features of the 3D-DWT / WSVC and against content-preserving operations. The sensitivity of the hash function to the content differences is also examined. The experiments are described below.

(a) Wavelet-based Scalable Coding

For testing the robustness of the hash function against the spatio-temporal scalabilities of the 3D-DWT / WSVC schemes, the following operations are carried out on each of the wavelet-decomposed original GOFs.

i. Temporal resolution reduction: The GOF can be temporarily scaled by dropping the high-pass frames above certain levels. Since the hash is extracted from the highest level of temporal

decomposition, the hash function is naturally robust against the temporal scaling in the 3D-DWT domain.

ii. Spatial resolution reduction: Spatial high-pass bands above certain levels can be dropped for the spatial scaling of the GOF. As the hash of the GOF is extracted from the $t5L - s3LL$ band, the high-pass bands up to the third level of spatial decomposition can be truncated without affecting the hash. In other words, the hash function is naturally robust against the spatial scaling of the GOF in the 3D-DWT domain up to the spatial dimension of $\frac{1}{64}$CIF.

iii. Bit-rate resolution reduction: The three quantizers used in the previous chapter to reduce the quality or bit-rate resolution are also used here. The wavelet coefficients in the $t5L - s3LL$ bands of the GOFs are quantized by applying the 8-bit, 7-bit, 6-bit and 5-bit quantizes separately. The resulting bands are used for examining the performance of the hash function against the bit-rate scalability.

Due to the natural robustness of the hash function against the temporal and spatial resolution reductions, the experimental results for the quantization operations are presented here. The hashes of the GOFs are computed after quantizing the wavelet coefficients in the $t5L - s3LL$ bands by applying the 8-bit, 7-bit, 6-bit and 5-bit quantizers separately.

(b) Content-Preserving Operations

The content-preserving operations considered in the previous chapter are also considered here. The hashes of the GOFs are computed after each operation.

i. MPEG compression: To examine the effect of the lossy compression on the contents of the representative frames, the GOFs are compressed using the MPEG-2 coder at the bit-rate of 64kbps and decompressed. The decompressed GOFs are considered for experimentation.

ii. Spatial averaging: Averaging masks of sizes 3×3 and 5×5 are applied on the frames of the GOFs for studying the effect of spatial averaging on the hash. This results in two processed GOFs from each original GOF.

iii. Brightness modification: For examining the robustness of the hash function against the brightness variations, intensity of each frame in the GOFs is increased/decreased by 50% of the original frame intensity.

iv. Contrast modification: The histogram equalisation method is used to enhance the contrast of the GOFs.

iv. Noise addition: Zero-mean AWGN with variance of 10 and 20 are added to the GOFs to test the resilience of the hash against the AWGN.

The quantization and content-preserving operations results in a group of 12 hashes of similar GOFs for each of the 56 original GOFs. The 12 hashes include the hash of the original GOF, the three hashes computed after quantizing the $t5L - s3LL$ band of the original GOF with each of the three quantizers and the eight hashes computed from GOFs obtained after the content-preserving operations on the original GOF. As mentioned earlier, an original GOF is equivalent to the result of performing the identity operation on the original GOF. Note that the result of the identity operation corresponds to 8-bit quantization.

(c) Content Differences

To examine the sensitivity of the proposed hash function to content differences, such differences at the block, frame and GOF levels are considered.

i. Block-level content difference: As done in the previous chapter, the performance of the hash function against the content differences at the block level is studied by modifying a block in the same position in each frame of the GOFs. Different sizes of blocks starting with the size of the frame are considered for experimentation. It is observed that the hash function can detect content modification up to a block size of about 5% of the frame size. The case of 5% block size is reported here.

   In each original GOF, another original GOF chosen at random is inserted in such a way that an inserted frame occupies 5% of the area of a frame. The 56 original GOFs results in 56 modified GOFs, where the corresponding GOFs have differences in their contents at the block level.

ii. Frame-level content difference: Different numbers of consecutive frames, starting from 32 frames per GOF, are considered for replacement. It is observed that the hash function can detect frame replacement up to 8 consecutive frames per GOF. The performance of the hash function against the replacement of 8 consecutive frames per GOF is reported here.

   To derive GOFs with frame-level differences in their contents, the last eight frames in each

original GOF are replaced with equal number of frames from another original GOF chosen at random. This derives another set of 56 modified GOFs.

iii. GOF-level content difference: To study the performance of the hash function in distinguishing content differences at the GOF level, the GOFs from the same and distinct videos are considered. For the 56 original GOFs, each pair of distinct GOFs represents content difference at the GOF level. Thus, $\frac{56\times55}{2} = 1540$ pairs of GOFs with dissimilar contents are possible.

In all the above cases, the similarity between two GOFs is determined by applying (5.8) and (5.10).

5.3.1 Average Performances

In this subsection, the average performances of the proposed hash function against the content-preserving operations and against the various types of content differences are presented.

(a) and (b) Quantization and Content-Preserving Operations

The average performances of the hash function against the identity, quantization and other content-preserving operations are examined here. The 56 groups of hashes, each containing the hashes of 12 similar GOFs, are considered. For example, in the case of the 7-bit quantization operation, the hashes derived after quantizing the $t5L - s3LL$ bands of the 56 original GOFs are compared with the corresponding hashes of the original GOFs by using (5.7). The 56 similarity values are computed by using (5.8). Thus, the average similarity value against the 7-bit quantization operation is the average of 56 similarity values. The average similarity values against each of the 12 operations are computed and shown in Figure 5.3 with the white bars. As expected, these values are close to one indicating the robustness of the hash function against the 12 operations.

(c) Content Differences

For computing the average performance against the differences in the contents of GOFs at the GOF-level, the similarity values are computed for dissimilar GOF pairs. For the identity operation, the similarity value for each original GOF and each of the other 55 original GOFs is computed by using (5.8). Hence, the average similarity value against the operation is the average of the $\frac{56\times55}{2} = 1540$ similarity values. Against each of the quantization and content-preserving operations, the average

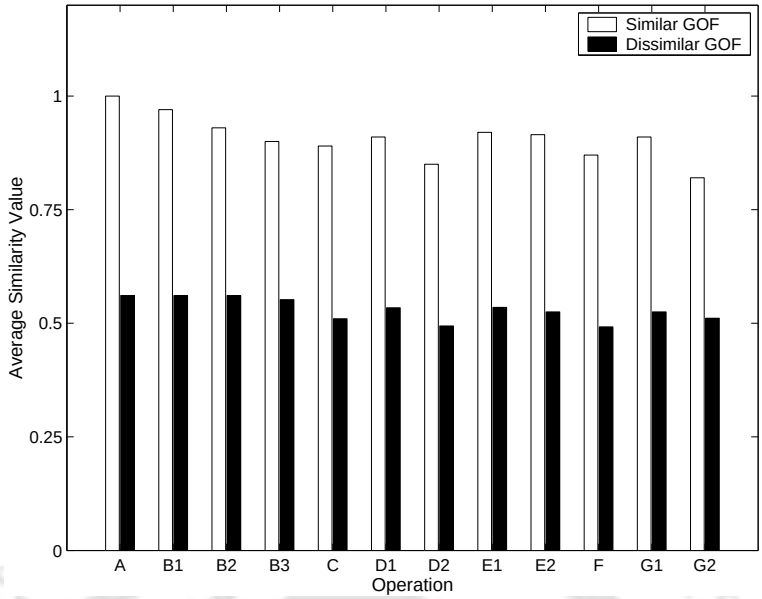


Fig. 5.3: The average performances of the hash function in terms of the average similarity values against the operations: (A) identity, (B) quantization (1) 7-bit (2) 6-bit (3) 5-bit, (C) MPEG-2 compression at the bit rate of 64kbps, (D) spatial averaging (1) 3×3 (2) 5×5, (E) brightness modification (1) +50% (2) -50%, (F) contrast modification (HE), (G) AWGN addition with variance (1) 10 (2) 20

similarity value is the average of $56 \times 55 = 3080$ similarity values. These average values against the various operations are also shown in Figure 5.3 with the black bars. These values are close to 0.5. Note that, this average similarity value 0.5 corresponds to the average Hamming distance 99 obtained from the statistical model in Subsection 5.1.2 for dissimilar GOFs. For similar contents, the observed minimum average similarity value is 0.82. The observed maximum average similarity value for dissimilar contents is 0.56. The well separation of the two values indicates that the hash function recognises similar and dissimilar GOFs well.

As mentioned earlier, the GOFs with block-level and frame-level content differences are derived in the same way those were derived in the previous chapter. In the case of block-level content difference (the case of 5% block size), the similarity value for each of the 56 original GOFs and the corresponding modified GOF with block-level content difference is computed by using (5.8) and the average similarity value is calculated. Similarly, the similarity value for each original GOF and the corresponding modified GOF with frame-level content difference (the case of replacement of 8 consecutive frames per GOF) is computed. The average of the 56 similarity values is calculated. The two average similarity values are presented in Figure 5.4. The average similarity value for the GOF-level content difference against the identity operation Figure 5.3 is also shown in Figure 5.4

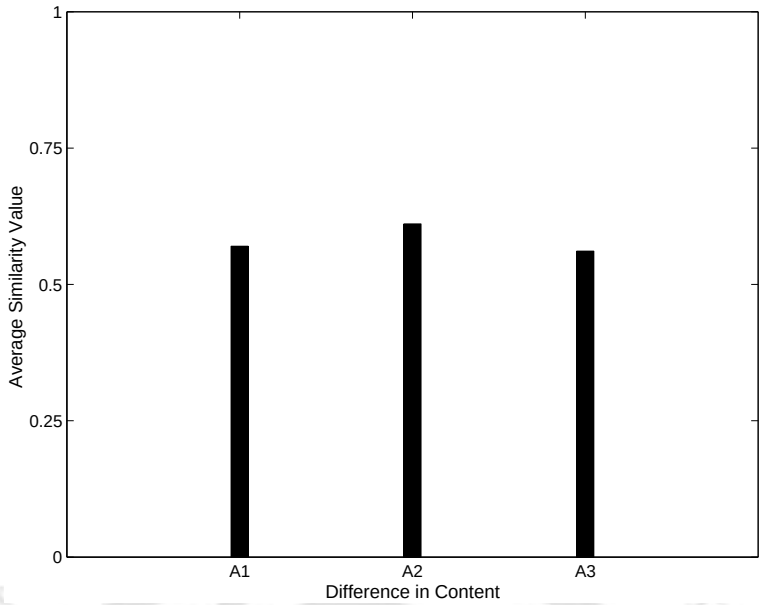


Fig. 5.4: The average performances of the hash function in terms of the average similarity values against the content differences at the: (A1) block level (for the case of 5% block size) (A2) frame level (for the case of replacement of 8 consecutive frames per GOF) (A3) GOF level

for comparison. As expected, the three values are close to 0.5 indicating that the proposed hash function is sensitive enough to detect content differences at the block, frame and GOF levels.

5.3.2 Diffusion and Confusion Properties

For good diffusion property, the hash function has to comprehend the differences in the contents of two distinct GOFs. To examine the diffusion property with the original GOFs, the hash of each original GOF is compared with the hashes of the other 55 original GOFs in terms of (5.7) and the similarity value is computed by using (5.8). Thus, the average similarity value is the average of the resulting $\frac{56\times55}{2} = 1540$ similarity values. The diffusion property is also examined against each of the quantization and content-preserving operations. Against an operation, the hash of each original GOF is compared with the hashes of the GOFs obtained after the operation on the other 55 GOFs. The average similarity value is the average of the resulting $56 \times 55 = 3080$ similarity values. A plot of the observed average similarity values is presented in Figure 5.5. The average similarity values close to 0.5 suggest good diffusion property of the hash function.

When two different secret keys K_1 and K_2 are used, the hash function should generate two distinct hashes from a GOF for good confusion property. The hash of each of the 56 original GOFs with the key K_1 is compared with the hash of the GOF with the key K_2 by using (5.7) and the similarity value is computed by using (5.8). The average of the resulting 56 similarity values is used to show

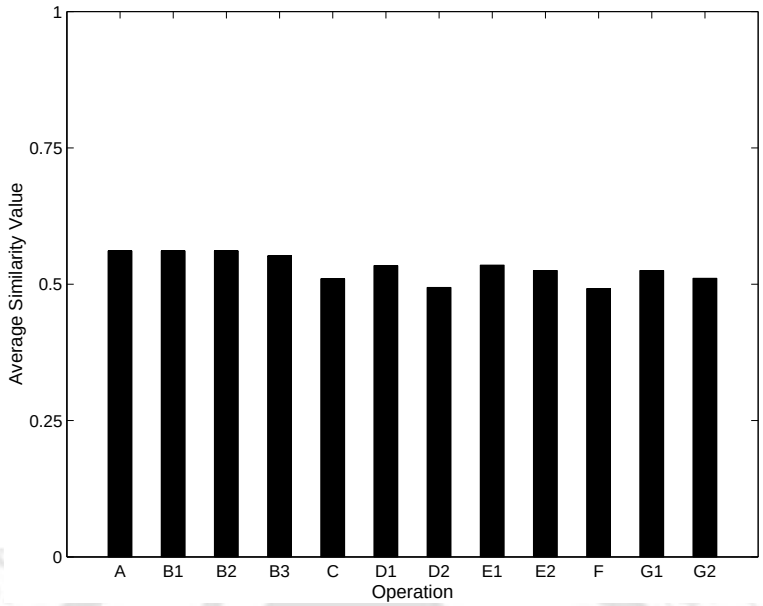


Fig. 5.5: The diffusion property of the hash function in terms of the average similarity values against the operations: (A) identity, (B) quantization (1) 7-bit (2) 6-bit (3) 5-bit, (C) MPEG-2 compression at the bit rate of 64kbps, (D) spatial averaging (1) 3×3 (2) 5×5, (E) brightness modification (1) +50% (2) -50%, (F) contrast modification (HE), (G) AWGN addition with variance (1) 10 (2) 20

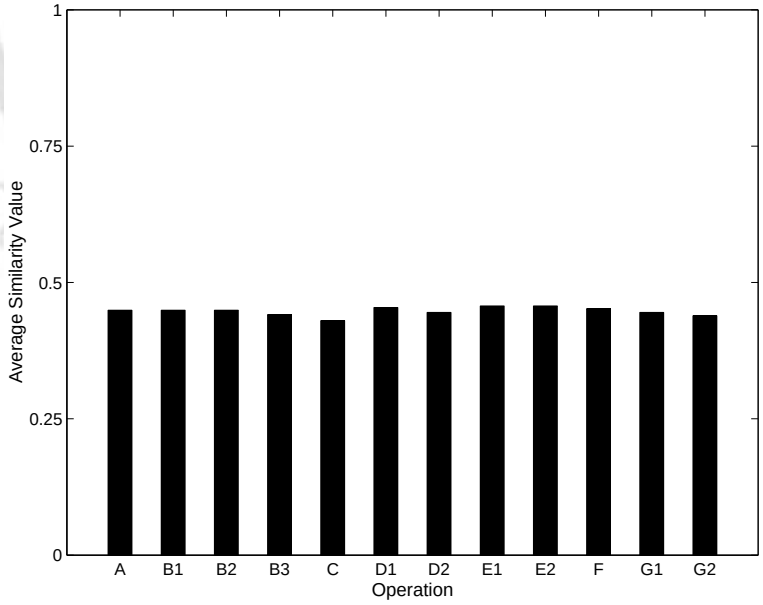


Fig. 5.6: The confusion property of the hash function in terms of the average similarity values against the operations: (A) identity, (B) quantization (1) 7-bit (2) 6-bit (3) 5-bit, (C) MPEG-2 compression at the bit rate of 64kbps, (D) spatial averaging (1) 3×3 (2) 5×5, (E) brightness modification (1) +50% (2) -50%, (F) contrast modification (HE), (G) AWGN addition with variance (1) 10 (2) 20

the confusion property of the hash function against the identity operation. It is also required that the quantization and content-preserving operations on GOFs should not affect this distinctiveness. Thus, the hash of each original GOF generated with K_1 and that of the output of an operation on the original GOF generated with K_2 are compared by using (5.7) and the similarity value is computed by applying (5.8). Against each operation, the average is computed over the 56 similarity values obtained with the 56 original GOFs. Figure 5.6 shows a plot of these values. These values close to 0.5 are indicative of the good confusion property of the hash function.

5.3.3 Experimental Verification of the Threshold

In Subsection 5.1.2, a value for the threshold T_6 was computed to be 0.75 based on the statistical model. We here evaluate a suitable value for T_6 from the experimental findings and compare it with the value obtained from the model. The 56 original GOFs are used in this experimentation. Each original GOF is passed through the quantization (except the 8-bit quantization) and content-preserving operations mentioned above and the hashes of the resulting GOFs are computed. These hashes are compared with the hash of the corresponding original GOF by using (5.7) and the similarity values are computed by using (5.8). Against each of the 11 operations, 56 similarity values are derived. Thus, a total of 616 similarity values are obtained. The histogram of the normalised frequencies of the 616 similarity values is shown in Figure 5.7(a).

Again, the similarity values for the all-possible pairs of the distinct original GOFs are computed. This results in 1540 similarity values. The histogram of the normalised frequencies of the similarity values for the distinct GOFs is shown in Figure 5.7(b).

It can be observed from the figure that a value for T_6 may be chosen in the range [0.625 0.875], which includes the value 0.75 obtained from the statistical model. Therefore, we use $T_6 = 0.75$ in the following to study the verification performance of the hash function.

5.3.4 Verification Performance

As discussed in the previous chapter, for a practical hash function, one likes to have the rates of false rejection and false acceptance very close to zero. During the analysis of the verification performance of the proposed hash function, we consider $T_6 = 0.75$. The percentages of FRR and FAR are computed against each of the identity, quantization and content-preserving operations similar to in Subsection 4.5.4. These rates are shown in Figure 5.8.

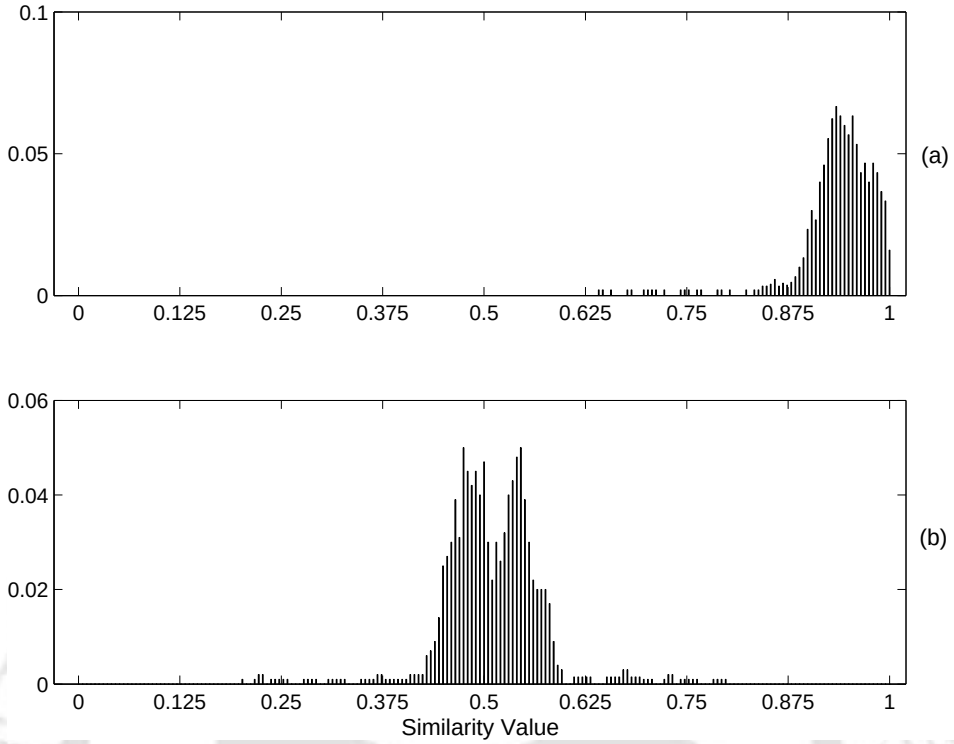


Fig. 5.7: The histograms of the normalised frequencies of the similarity values: (a) similar GOFs (b) dissimilar GOFs

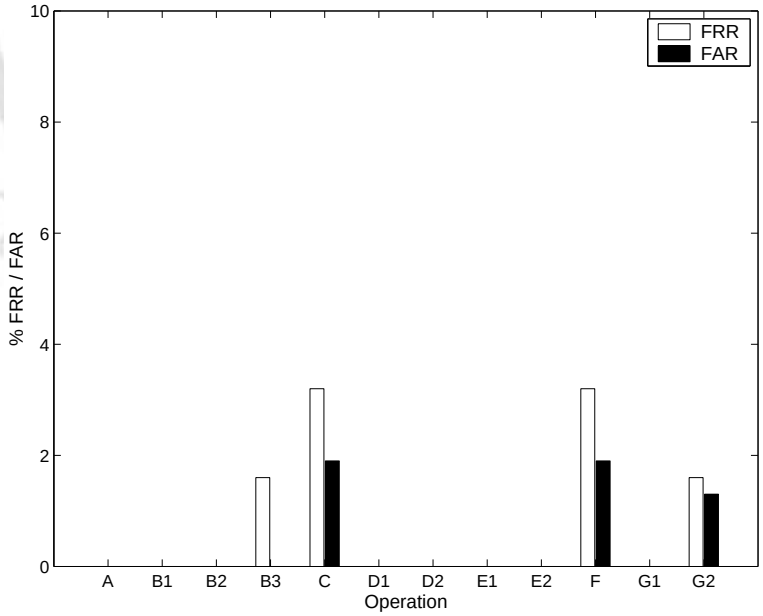


Fig. 5.8: The FRR and FAR rates against the operations: (A) identity, (B) quantization (1) 7-bit (2) 6-bit (3) 5-bit, (C) MPEG-2 compression at the bit rate of 64kbps, (D) spatial averaging (1) 3×3 (2) 5×5, (E) brightness modification (1) +50% (2) -50%, (F) contrast modification (HE), (G) AWGN addition with variance (1) 10 (2) 20

It can be seen in the figure that the FRR is non-zero when a coarse quantizer is selected during the bit-rate scaling (B2). The FRR and FAR rates are non-zero against the MPEG-2 compression at the bit rate of 64 kbps (C) and against the contrast modification (F). The non-zero FRR and FAR rates in the case of the MPEG-2 compression is due to the high distortion at a low bit-rate like 64 kbps. In the case of the contrast modification, the non-zero rates may be due to the changes in the spatio-temporal low-pass content of the GOFs introduced by the histogram equalisation. The non-zero FRR and FAR rates are also observed in the case of the AWGN addition with $\sigma^2 = 20$ (G2). This indicates that AWGN with high variances affect the performance of the hash function.

5.4 Discussion

In this chapter, a new perceptual hash function for 3D-DWT coded video has been presented. It derives a hash of a video at the GOF level from the spatio-temporal low-pass band at the full level of temporal and an intermediate level of spatial decomposition. The robustness of the hash function against the 3D-DWT based scalabilities and the common content-preserving operations has been examined. Further, the sensitivity to the differences in the contents at various levels has been verified. Experimental results demonstrate the effectiveness of the hash function. The hash function is found to have good diffusion and confusion properties. The observed FRR and FAR rates against various content-preserving operations are nominal. The low hash size, the robustness against the content-preserving operations, the sensitivity to the content differences and good diffusion and confusion properties make the hash function suitable for practical applications.

Chapter 6

Conclusions

The 3D-DWT based video coding promises to be an alternative to the conventional hybrid DCT-based video coding standards because of its attractive features like the inherent spatio-temporal scalabilities. A WSVC scheme can accommodate the temporal scalability, spatial scalability and bit-rate scalability by effective exploitation of the multi-resolution property of the 3D-DWT. The thesis explored video representation and hashing at the GOF level by using the bands of 3D-DWT with an aim to their use in the WSVC framework. The main contributions of the thesis are summarised in Section 6.1 and a few directions for future research are outlined in Section 6.2.

6.1 Summary of Contributions

This thesis addressed two problems related to the video representation and hashing:

i. Representation of the content of a video by the spatio-temporal bands derived from the 3D-DWT decomposition at the GOF level.

ii. Designing of perceptual hash functions from the perceptually-representative spatio-temporal band with robustness against the scalability features of the 3D-DWT based scalable coding and other content-preserving operations and sensitivity to content differences.

The first problem addressed by the thesis is how to represent the content a GOF in the 3D-DWT domain by means of representative bands. Chapter 3 explored the bands of the 3D-DWT decomposition of a video for representation. The contributions of this chapter are as follows:

i. The low-pass and high-pass bands of temporal wavelet decomposition of GOFs were fist examined for content representation. The results of detailed experimentations were presented to analyse the performance of various temporal bands in representing the GOFs. It was observed

that the temporal low-pass band at the full level of the decomposition can be used for the representation of the content of a GOF.

ii. For a more compact representation, the spatio-temporal low-pass bands of the 3D-DWT decomposition of the GOFs were examined. Experiments were performed to demonstrate the performances of these bands at different levels of decomposition. The spatio-temporal low-pass band at the full-level of temporal and at an intermediate level of spatial decomposition of a GOF was proposed for representing the content of a GOF.

iii. A novel similarity measure was proposed. It compares two representative frames based on the binarisation of their their local contents in perceptual blocks and finding the maximum of the Hamming distances between corresponding binarised perceptual blocks. A statistical model for the proposed similarity measure was also presented.

The second problem addressed by the thesis is the hashing of videos in the WSVC framework such that the hashes are robust against the scalabilities of the WSVC schemes. Chapter 4 proposed a perceptual hash function for video in the 3D-DWT domain. The hash function computes a hash of a video at the GOF level. The spatio-temporal low-pass band of a GOF, which was proposed for representation in the previous chapter, is divided into perceptual blocks. A hash of the GOF is extracted by thresholding the wavelet coefficients in each block about a local mean computed for the block. As the hash is computed from the spatio-temporal low-pass band, it is naturally robust against the spatio-temporal scalabilities of the WSVC schemes. The novel similarity measure in the previous chapter was also used here in hash comparison. Experiments were performed to study the performance of the hash function. The hash function showed good robustness against the quantization and other content-preserving operations. The robustness against the quantization ensures the robustness against the bit-rate scalability feature of a WSVC scheme. It also showed good sensitivity to content differences in distinct GOFs. However, the hash function has the following drawbacks:

i. The size of the hash is comparatively large.

ii. The diffusion and confusion properties of the hash function are weak.

Despite the weak diffusion property, the hash function showed good sensitivity to content differences in dissimilar GOFs due to the novel similarity measure.

In Chapter 5, the drawbacks of the perceptual hash function proposed in the previous chapter were addressed and a new perceptual hash function for the wavelet-coded video was proposed. This hash function computes a hash of a GOF from the spatio-temporal low-pass band like in the previous method. This band is divided into perceptual blocks and the content of each block is represented by the local mean. The forward and backward cumulative averages of these local means are binarised to extract the hash. A very compact hash is extracted by this method. Detailed experimentation showed the robustness of the hash function against quantization and other content-preserving operations. The hash function was found to have high sensitivity to content differences. It also has strong diffusion and confusion properties, thereby demonstrating the workability of the hash.

6.2 Future Research Directions

The proposed solutions are novel attempts for perceptual hashing of video coded in the WSVC framework. This work points to new research directions. Some of these are outlined below.

i. *Representation and hashing using the spatio-temporal bands of motion-compensated (MC) 3D-DWT*: A generalisation of the conventional DWT to the temporal dimension for video coding has limitations like coding delay and frame-memory cost due to the use of long temporal filters. Loss in compression is another drawback due to the significant energy in the high-pass bands. Moreover, temporal filtering yields a blurred low-pass band. Therefore, a better approach is to use the MC 3D-DWT. To avoid the computational complexity involved in the motion estimation, we considered in this thesis the simple temporal DWT and observed good results for the proposed representation and hashing solutions. The future work may study the performances of the proposed solutions in the MC 3D-DWT domain.

ii. *Robustness against geometric operations*: The proposed solutions in this thesis are not robust against the geometric operations like rotation, shearing, cropping, etc. on video. Hash functions in the 3D-DWT domain with robustness also against there operations are to be designed.

iii. *Robustness of the proposed perceptual hash functions against SVC schemes based on MPEG-X and H.26X*: The proposed hash functions are found to be robust against MPEG-2 based coding. The performances of the hash functions against MPEG-x and H.26x may be studied.

iv. *Representation and hashing of colour video*: We considered the luminance component of a video for content representation and for hashing. The information in the colour spaces are

also fundamental. The proposed solutions for video content representation and video hashing might be extendable to include the colour information. Performance may be studied when the luminance component along with the colour information or the colour information alone are considered.

v. *Performance testing using a standardised WSVC scheme*: As a standerdised WSVC system is yet to become a reality, we employed the Haar wavelets for temporal decomposition and the Daubechies 9/7 biorthogonal wavelets for spatial decomposition for their widespread uses. We also examined the performances of the proposed solutions against the bit-rate scalability of WSVC by using quantizers. Once a WSVC system is standardised, the performance analysis for each solution is to be made based on the standard.

vi. *Segment handling in the video identification application*: The proposed content representation solution considers 3D-DWT coded video at the GOF level. It is useful for identifying a GOF in a video. But, identification of such a short sequence of frames in video may not be always necessary. Although a video segment can be represented by the representative frames of the GOFs, the involved memory requirement and the computational complexity in the similarity determination may not be always acceptable. Hence, the optimal representation of video segments in the 3D-DWT domain may be studied.

vii. *Hierarchical Hashing of video*: The proposed hash functions compute video hashes at the GOF level so that the contents of videos can be verified and localised at the GOF level. The verification of the hash may not be always necessary at the GOF level. To avoid the unnecessary time complexity in those situations, hierarchical hashing solutions may be proposed in the WSVC framework.

viii. *Representation and hashing of JPEG 2000 image*: The proposed representation and hashing solutions might be also applicable to JPEG2000 images. The spatial low-pass band at an intermediate level of wavelet decomposition of an image may be used for representing the image. Hashes for the image may be computed from this band by using the two proposed hash functions. The performances of the representation and the hash functions for JPEG 2000 images may be examined.

Bibliography

[1] J. W. Rittinghouse and W. M. Hancock, *Cybersecurity Operations Handbook*. MA: Digital Press, 2003.

[2] C. Kaufman, R. Perlman and M. Speciner, *Network Security: Private Communication in a Public World*, Singapore: Pearson Education, 2005.

[3] V. Monga and B. L. Evans, "Robust perceptual image hashing using feature points," in *Proceedings of the IEEE International Conference on Image Processing*, Singapore, Oct. 2004, vol. 1, pp. 677-680.

[4] V. Monga and B. L. Evans, "Perceptual Hashing via Image Feature Points: Performance Evaluation and Trade-offs," *IEEE Transactions on Image Processing*, vol. 15, no. 11, pp. 3452-3465, 2006.

[5] S.-H. Han, C.-H. Chu and S. Yang, "Content-based Image Authentication: Current Status, Issues and Challenges," n *Proceedings of the IEEE International Conference on Semantic Computing*, Irvine, CA, Sept. 2007, vol. 1, pp. 630-636.

[6] S.-H. Han and C.-H. Chu, "Content-based Image Authentication: Current Status, Issues and Challenges," *Springer International Journal of Information Security*, vol. 9, no. 1, pp. 19-32, Feb. 2010.

[7] S.-H. Han, C.-H. Chu and S. Yang, "Content-based Image Authentication: Current Status, Issues and Challenges," *ACM International Journal of Information Security*, vol. 9, no. 1, pp. 19-32, Jan. 2010.

[8] J. Oostveen, T. Kalker and J. Haitsma, "Visual Hashing of Digital Video: Applications and Techniques," in *Applications of Digital Image Processing Conference XXIV, Proceedings of SPIE*, San Diego, CA, Jul./Aug. 2001, vol. 4472, pp. 121-131.

[9] Y. Tao, V. Muthukkumarasamy, B. Verma and M. Blumenstein, "A Texture Feature Extraction Technique Using 2D-DFT and Hamming Distance" in *Proceedings of the IEEE International*

Conference on Computational Intelligence and Multimedia Applications, Xían, China, Sept. 2003, pp. 120-125.

[10] D. P. Huttenlocher, G. A. Klanderman and W. J. Rucklidge, "Comparing Images Using the Hausdorff Distance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 9, pp. 850-863, Sep. 1993.

[11] M.-P. Dubuisson and A. K. Jain, "A Modified Hausdorff Distance for Object Matching," in *Proceedings of the IEEE International Conference on Pattern Recognition*, Jerusalem, Israel, Oct. 1994, pp. 566-568.

[12] B. Coskun, and N. Memon, "Confusion/ Diffusion Capabilities of Some Robust Hash Functions," in *Proceedings of the IEEE International Conference on Information Sciences and Systems*, Princeton, NJ, Mar. 2006, pp. 1188-1193.

[13] T. C. Hoad and J. Zobel, "Fast Video Matching with Signature Alignment," in *Proceedings of the ACM SIGMM International Workshop on Multimedia Information Retrieval*, Berkeley, CA, Nov. 2003, pp. 262-269.

[14] A. Mucedero, R. Lancini and F. Mapelli, "A Novel Hashing Algorithm for Video Sequences," in *Proceedings of the IEEE International Conference on Image Processing*, Singapore, Oct. 2004, vol. 4, pp. 2239-2242.

[15] R. Lancini, F. Mapelli and A. Mucedero, "Automatic identification of Compressed Video," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Quebec, Canada, May 2004, vol. 3, pp. 445-448.

[16] Y.-N. Li and Z.-M. Lu, "Video Identification Using Spatio-Temporal Salient Points," in *Proceedings of the IEEE International Conference on Information Assurance and Security*, Xían, China, Aug. 2009, vol. 2, pp. 79-82.

[17] T. Kalker, J. Haitsma and J. Oostveen, "Issues with Digital Watermarking and Perceptual Hashing", in *Multimedia Systems and Applications Conference IV, Proceedings of SPIE*, Denver, CO, Aug. 2001, vol. 4518, pp. 189-197.

[18] M. Bober and P. Brasnell, "MPEG7 Visual Signature Tool," in *Proceedings of the IEEE International Conference on Multimedia and Expo*, New York, NY, June-July. 2009, pp. 1540-1543.

[19] J.-R. Ohm, "Advances in Scalable Video Coding," *Proceedings of IEEE*, vol. 93, no. 1, pp. 42-56, Jan. 2005.

[20] A. Vetro, C. Christopoulos, and H. Sun, "Video Transcoding Architectures and Techniques: An Overview," *IEEE Signal Processing Magazine*, vol. 20, no. 2, pp. 18-29, Mar. 2003.

[21] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103-1120, Sep. 2007.

[22] N. Adami, A. Signoroni, and R. Leonardi, "State-of-the-Art and Trends in Scalable Video Compression With Wavelet-Based Approaches," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1238-1255, Sep. 2007.

[23] K.-T. Fung, Y.-L. Chan and W.-C. Siu, "New Architecture for Dynamic Frame-Skipping Transcoder," *IEEE Transactions on Image Processing*, vol. 11, no. 8, pp. 886-900, Aug. 2002.

[24] W. Zhu, K. H. Yang and M. J. Beacken, "CIF-to-QCIF Video Bitstream Down-Conversion in the DCT Domain," *Bell Labs Technical Journal*, vol. 3, no. 3, pp. 21-29, Jul.-Sep. 1998.

[25] H. Sun, W. Kwok and J. W. Zdeoski, "Architectures for MPEG Compressed Bitstream Scaling," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 2, pp. 191-199, Mar. 1996.

[26] "Coding of Audio-video objects Part-2: Visual," ISO/IEC 14492-2 (MPEG-4 Visual), ISO/IEC JCT 1, Version 1: Apr. 1999, Version 2: Feb. 2000, Version 3, May 2004.

[27] "Advance video coding for generic audiovisual services," ITU-T Rec. H.264, ISO/IEC 14496-10 (MPEG-4 AVC), ITU-T and ISO/IEC JCT 1, Version 1: May 2003, Version 2: May 2004, Version 3: Mar. 2005, Version 4: Sept 2005, Version 5 and Version 6: Jun. 2006, Version 7: Apr. 2007, Version 8 (including SVC extension): Consented in Jul. 2007.

[28] "Joint draft 7 of SVC amendment (Revision 2)," ITU-T and ISO/IEC JCT1, JVT-T201r2, Jul. 2006.

[29] "Joint scalable video model JSVM-6," ITU-T and ISO/IEC JCT1, JVT-S202, Apr. 2006.

[30] "Joint Scalable Video Model (JSVM) 6.0," Montreux, Switzerland, ISO/MPEG Video, Tech. Rep. N8015, Apr. 2006.

[31] R. Xiong, J. Xu, F. Wu, S. Li and Y.-Q. Zhang, "Subband Coupling Aware Rate Allocation for Spatial Scalability in 3-D Wavelet Video Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 10, pp. 1311-1324, Oct. 2007.

[32] S.-T. Hsiang, *Highly Scalable Subband/Wavelet Image and Video Coding*, PhD Thesis, Rensselaer Polytechnic Institute, Troy, New York, Jan. 2002.

[33] B-J. Kim, Z. Xiong and W. A. Pearlman, "Low Bit-Rate Scalable Video Coding With 3-D Set Partitioning in Hierarchical Trees (3-D SPIHT)," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 8, pp. 1374-1387, Dec. 2000.

[34] D. Athanasopoulos and T. Stouraitis, "Content-Adaptive Wavelet-Based Scalable Video Coding," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, New Orleans, LA, May. 2007, pp. 981-984.

[35] R. Leonardi, A. Signoroni, S. Brangoulo, "Status report-version 1 on wavelet video coding exploration," ISO/IEC JTC1/SC29/WG11, ISO/MPEG Video, Tech. Rep. N7822, Jan. 2006.

[36] R. M. Rao and A. S. Bopardikar, *Wavelet Transforms: Introduction to Theory and Applications*. MA: Addison-Wesley, 1998.

[37] G. Strang and T. Q. Nguyen, *Wavelets and Filter Banks*. MA: Wellesley-Cambridge Press, 1996.

[38] J. Shapiro, "Embedded Image Coding Using Zerotress of Wavelet Coefficients," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3445-3462, Dec. 1993.

[39] A. Said and W. A. Pearlman, "A New, Fast, and Eefiicient Image Codec Based on Set Partitioning in Hierarchical Trees," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 243-250, Jun. 1996.

[40] S.-T. Hsiang and J. W. Woods, "Embedded image coding using zeroblocks of subband/wavelet coefficients and context modeling," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, Geneva, Switzerland, May 2000, vol. 3, pp. 662-665.

[41] F. Lazzaroni, A. Signoroni and R. Leonardi, "Embedded morphological dilation coding for 2-D and 3-D images," in *Visual Communications and Image Processing Conference, Proceedings of SPIE*, San José, CA, Jan. 2002, vol. 4671, pp. 923-934.

[42] F. Lazzaroni, R. Leonardi and A. Signoroni, "High-Performance Embedded Morphological Wavelet Coding," *IEEE Signal Processing Letters*, vol. 10, no. 10, pp. 293-295, Oct. 2003.

[43] D. Taubman, E. Ordentlich, M. Weinberger and G. Seroussi, "Embedded block coding in JPEG 2000," *Signal Processing : Image Communication*, vol. 17, pp. 4972, Jan. 2002.

[44] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Transactions on Image Processing*, vol. 9, no. 7, pp. 1158-1170, Jul. 2000.

[45] S.-T. Hsiang and J. Woods, "Embedded Video Coding Using Invertible Motion Compensated 3-D Subband/Wavelet Filter Bank," *Signal Processing : Image Communication*, vol. 16, no. 8, pp. 705724, May 2001.

[46] P. Chen and J. W. Woods, "Bidirectional MC-EZBC with lifting implementation, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 10, pp. 11831194, Oct. 2004.

[47] N. Adami, M. Brescianini, M. Dalai, R. Leonardi and A. Signoroni, "A Fully Scalable Video Coder with Inter-Scale Wavelet Prediction and Morphological Coding," in *Visual Communications and Image Processing Conference, Proceedings of SPIE*, Beijing, China, Jul. 2005, vol. 5960, pp. 535-546.

[48] "Wavelet Codec Reference Document and Software Manual," ISO/IEC JTC1/SC29/WG11, 73^{th} MPEG Meeting, Poznan, Poland, ISO/MPEG Video, Tech. Rep. N7334, Jul. 2005.

[49] C. I. Podilchuk, N. S. Jayant and N. Farvardin, "Three-dimensional Subband Coding of Video," *IEEE Transactions on Image Processing*, vol. 4, no. 2, pp. 125-139, Feb. 1995.

[50] X. Yang and K. Ramchandran, "Scalable Wavelet Video Coding Using Aliasing-Reduced Hierarchical Motion Compensation," *IEEE Transactions on Image Processing*, vol. 9, no. 5, pp. 778-791, May 2000.

[51] V. Bottreau, M. Bénetière, B. Felts and B. Pesquet-Popescu, "A Fully Scalable 3-D Subband Video Codec," in *Proceedings of the IEEE International Conference on Image Processing*, Thessaloniki, Greece, Oct. 2001, vol. 2, pp. 1017-1020.

[52] H. Danyali and A. Mertins, "A Fully SNR, Spatial and Temporal Scalable 3DSPIHT-Based Video Coding Algorithm for Video Streaming Over Heterogeneous Networks," in *Proceedings of the IEEE International Conference on Convergent Technologies for Asia-Pacific Region*, Bangalore, India, Oct. 2003, vol. 4, pp. 1445-1449.

[53] S. Yea and W. A. Pearlman, "A Wavelet-Based Two-Stage Near-lossless Coder," *IEEE Transactions on Image Processing*, vol. 15, no. 11, pp. 3488-3500, Nov. 2006.

[54] Q. Sun and S-F. Chang, "A Secure and Robust Digital Signature Scheme for JPEG 2000 Image Authentication System," *IEEE Transactions on Multimedia*, vol. 7, no. 3, pp. 480-494, Jun. 2005.

[55] Q. Sun, D. He and Q. Tian, "A Secure and Robust Authentication Scheme for Video Transcoding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 10, pp. 1232-1244, Oct. 2006.

[56] B. Coskun, B. Sankur and N. Memon, "Spatio-Temporal Transform Based Video Hashing," *IEEE Transactions on Multimedia*, vol. 8, no. 6, pp. 1190-1208, Dec. 2006.

[57] M. Malekesmaeili, M. Fatourechi and R. K. Ward, "Video Copy Detection Using Temporally Informative Representative Images," in *Proceedings of the IEEE International Conference on Machine Learning and Applications*, Miami Beach, Florida, Dec. 2009, pp. 69-74.

[58] C. D. Roover, C. D. Vleeschouwer, F. Lefébvre, and B. Macq, "Robust Video Hashing Based on Radial Projections of Key Frames," *IEEE Transactions on Signal Processing*, vol. 53, no. 10, pp. 4020-4037, Oct. 2005.

[59] M. Schneider and S.F. Chang, "A Robust Content Based Digital Signature for Image Authentication," in *Proceedings of the IEEE International Conference on Image Processing*, Lausanne, Switzerland, Sept. 1996, vol. 3, pp. 227-230.

[60] M. Alghoniemy and A.H. Tewfik, "Geometric Invariance in Image Watermarking," *IEEE Transactions on Image Processing*, vol. 13, no. 2, pp. 145-153, Feb. 2004.

[61] J. Dittman, A. Steinmetz and R. Steinmetz, "Content-based Digital Signature for Motion Pictures Authentication and Content-Fragile Watermarking," in *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, Florence, Italy, Jun. 1999, vol. 2, pp. 209-213.

[62] S.J.. Xiang, H.J. Kim and J.W. Huang, "Histogram-based Image Hashing Scheme Robust Against Geometric Deformation," in *Proceedings of the ACM Workshop on Multimedia & Security*, Dallas, Texas, Sept. 2007, pp. 121-128.

[63] R. Venkatesan, S.-M. Koon, M.-H. Jakubowski, and P. Moulin, "Robust Image Hashing," in *Proceedings of the IEEE International Conference on Image Processing*, Vancouver, BC, Canada, Sept. 2000, vol. 3, pp. 664-666.

[64] S. Bhattacherjee and M. Kutter, "Compression Tolerant Image Authentication," in *Proceedings of the IEEE International Conference on Image Processing*, Chicago, IL, Oct. 1998, vol. 1, pp. 435-439.

[65] E.-C. Chang, M. S. Kankanhalli, X. Guan, Z. Huang and Y. Wu, "Robust Image Athentication Using Content Based Compression," *ACM/Springer International Journal of Multimedia Systems*, vol. 9, no. 2, pp. 121-130, Aug. 2003.

[66] C.-Y. Lin and S.-F. Chang, "A Robust Image Authentication Method Distinguishing JPEG Compression from Malicious Malicious Manipulation," *IEEE Transactionson Circuits and Systems for Video Technology*, vol. 11, no. 2, pp. 153-168, Feb. 2001.

[67] C.-Y. Lin and S.-F.Chang, "Robust Image Authentication Method Surviving JPEG Lossy Compression," in *Storage and Retrieval for Image and Video Databases Conference VI, Proceedings of SPIE*, San José, CA, Jan. 1998, vol. 3312, pp. 296-307.

[68] C.-Y. Lin and S.-F.Chang, "An Image Authenticator Surviving DCT-based Variable Quantization Table Compressions," CU/CTR, New York, Tech. Rep. 490-98-24, Nov. 1997.

[69] C.-S. Lu and H.-Y. M. Liao, "Structural Digital Signature for Image Authentication: An Incidental Distortion Resistant Scheme," *IEEE Transactions on Multimedia*, vol. 5, no. 2, pp. 161-173, Jun. 2003.

[70] J. Fridrich, "Robust Bit Extraction from Images," in *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, Florence, Italy, Jun. 1999, vol 2, pp. 536-540.

[71] J. Fridrich and M. Goljan, "Robust Hash Functions for Digital Watermarking," in *Proceedings of the IEEE International Conference on Information Technology: Coding and Computing*, Las Vegas, NA, Mar. 2000, pp. 178-183.

[72] M.K. Mihcak and R. Venkatesan, "New Iterative Geometric Methods for Robust Perceptual Image Hashing," in *Proceedings of the ACM CCS-8 Workshop on Security and Privacy in Digital Rights Management*, Philadelphia, PA, Nov. 2001, pp. 13-21.

[73] A. Swaminathan, Y. Mao and M. Wu, "Robust and Secure Image Hashing," *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 2, pp. 215-230, Jun. 2006.

[74] T. Uehara, R. Safavi-Naini and P. Ogunbona, "A Secure and Flexible Athentication System for Digital Images," *ACM/Springer International Journal of Multimedia Systems*, vol. 9, no. 5, pp. 441-456, Mar. 2004.

[75] F. Ahmed and M. Y. Siyal, "A Secure and Robust Wavelet-Based Hashing Scheme for Image Authentication," in *Proceedings of the International Multimedia Modeling Conference*, Singapore, Vol. 2, Jan. 2007, pp. 51-62.

[76] A. Nagasaka and Y. Tanaka, "Autometic Video Indexing and Full-Video Search for Object Appearances," in *Proceedings of the IFIP TC2/WG 2.6 Second Working Conference Visual Database Systems II*, Budapest, Hungary, sept.-Oct. 1991, pp. 113-127.

[77] W. Wolf, "Key Frame Selection by Motion Analysis," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Atlanta, GA, May 1996, vol 2, pp. 1228-1231.

[78] R. Radhakrishnan and C. Bauer, "Content-based Video Signatures based on Projections of Difference Images," in *Proceedings of the IEEE Workshop on Multimedia Signal Processing*, Crete, Oct. 2007, pp. 341-344.

[79] P. K. Atrey, W. Q. Yan, and M. S. Kankanhalli, "A Scalable Signature Scheme for Video Authentication," *Springer International Journal of Multimedia Tools and Applications*, vol. 34, no. 1, pp. 107-135, July 2007.

[80] S. Lee and C. D. Yoo, "Robust Video Fingerprinting for Content-Based Video Identification," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 7, pp. 983-988, Jul. 2008.

[81] A. Shivadas and J. M. Gauch, "Real-Time Commercial Recognition Using Color Moments and Hashing," in *Proceedings of the Fourth Canadian Conference on Computer and Robot Vision*, Montreal, Que, May 2007, pp. 465-472.

[82] F. Ahmed and M. Y. Siyal, "A Robust and Secure Signature Scheme for Video Authentication," in *Proceedings of the IEEE International Conference on Multimedia and Expo*, Beijing, China, Jul. 2007, pp. 2126-2129.

[83] T. Uehara, R. Safavi-Naini and P. Ogunbona, "An MPEG Tolerant Authentication System for Video Data," in *Proceedings of the IEEE International Conference on Multimedia and Expo*, Taipei, Taiwan, Jun. 2004, pp. 891-894.

[84] D. He, Q. Sun and Q. Tian, "A Semi-Fragile Object Based Video Authentication System," in *Proceedings of the IEEE International Symposium on Circuits and Systems)*, Bangkok, Thailand, May. 2003, vol. 3, pp. 814-817.

[85] M. S. Lew, N. Sebe and C. Gardner, "Video Indexing and Understanding" in *Principles of Visual Information Retrieval*, Berlin: Springer, pp. 163-196, 2001.

[86] X. Gao, H. Xin, and H. Ji, "A Study of Intelligent Video Indexing System," in *Proceedings of the IEEE Fourth World Congress on Intelligent Control and Automation*, Shanghai, China, Jun. 2002, vol. 3, pp. 2122-2126.

[87] S. Panchanathan, M.K. Mandal and T. Aboulnasr, "Video Indexing in the Wavelet Compressed Domain," in *Proceedings of the IEEE International Conference on Image Processing*, Chicago, Illinois, Oct. 1998, vol. 3, pp. 546-550.

[88] E. Ardizzone, M. La Cascia and D. Molinelli, "Motion and Color Based Video Indexing and Retrieval," in *Proceedings of the IEEE International Conference on Pattern Recognition*, Vienna, Austria, Aug. 1996, vol. 3, pp. 135-138.

[89] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*. McGraw Hill, New York, 1991.

[90] C.-Y. Tsai and H.-M. Hang, "ρ-GDD Source Modeling for Wavelet Coefficients in Image/Video Coding," in *Proceedings of the IEEE International Conference on Multimedia and Expo*, Hannover, Germany, Jun. 2008, pp. 601-604.

[91] R. P. Jain, *Modern Digital Electronics*. McGraw-Hill, USA, 2006.

[92] A. K. Jain, *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[93] M. Antonini, M. Barlaud, P. Mathieu and I. Daubechies, "Image Coding Using Wavelet Transform," *IEEE Transactions on Image Processing*, vol. 1, no. 2, pp. 205-220, 1992.

[94] M. Unser and T. Blu, "Mathematical Properties of the JPEG2000 Wavelet Filters," *IEEE Transactions on Image Processing*, vol. 12, no. 9, pp. 1080-1090, Sept. 2003.

[95] S.-J. Choi and J. W. Woods, "Motion-compensated 3-D Subband Coding of Video," *IEEE Transactions on Image Processing*, vol. 8, no. 2, pp. 155-167, Feb. 1999.

[96] J.-R. Ohm, "Three-Dimensional Subband Coding with Motion Compensation," *IEEE Transactions on Image Processing*, vol. 3, no. 5, pp. 559-571, Sept. 1994.

[97] Y. Liu, F. Wu and K. N. Ngan, "3-D Object-Based Scalable Wavelet Video Coding With Boundary Effect Suppression," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 5, pp. 639-644, May 2007.

[98] D. LeGall and A. Tabatabai, "Subband Coding of Digital Images Using Symmetric Short Kernel Filters and Arithmetic Coding Techniques," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, New York, NY, Apr. 1988, vol. 3, pp. 761-764.

[99] D. Taubman, "High Performance Scalable Image Compression with EBCOT," *IEEE Transactions on Image Processing*, vol. 9, no. 7, pp. 1158-1170, Jul. 2000.

[100] S. Thiemert, H. Sahbi and M. Steinebach, "Using Entropy for Image and Video Authentication Watermarks" in *Security, Steganography, and Watermarking of Multimedia Contents Conference VIII, Proceedings of SPIE*, San José, CA, Feb. 2006, 6072, pp. 607218.

[101] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1996.